LP Characteristic Vector of Logic Functions

Norio Koda

Department of Computer Science
and Electronic Engineering
Tokuyama College of Technology
Tokuyama, 745 Japan

Tsutomu Sasao

Department of Computer Science
and Electronics
Kyushu Institute of Technology
Iizuka, 820 Japan

Abstract

In this paper, we define the LP equivalence relation, and introduce an LP characteristic vector, which is invariant under this relation. We derive the characteristic vectors for functions on five or fewer input variables and show that the characteristic vectors uniquely identify the equivalence classes. We have obtained the table of minimum AND-EXOR expressions (MESOPs) for the representative functions of the LP equivalence class of 5 variables. Thus, the MESOP of a given 5-variable function can be found by a table look-up method.

1 Introduction

With the increasing complexity of LSI, logic design has become the work of a logic synthesis system instead of a human. Most logic synthesis tools use AND and OR gates as basic logic elements. However, arithmetic and error correcting circuits can be realized with many fewer gates if EXOR gates are available as well as AND and OR gates. Table 1 compares the number of 4-variable fucntions requiring t products. This shows that AND-EXORs require fewer products than AND-ORs. Thus, the establishment of a design method using EXOR gates in addition to AND, OR and NOT gates is vitally important.

Among various AND-EXOR type logical expressions[1,9], ESOP are the most general class and can represent functions with the fewest products[7]. But, the minimization of ESOPs is extremely difficult. Although minimum solutions can be obtained by an exhaustive or a virtually exhaustive method for functions with a small number of variables[4,6], an exact minimization of ESOPs is difficult in most cases. Therefore, heuristic algorithms to obtain near minimum ESOPs have been developed[3,10].

When an ESOP of an n-variable function is expanded by one or more variables, ESOPs of sub-functions with $n-1$ or fewer variables are obtained. If we know minimum ESOPs (MESOPs) for these sub-functions, we can obtain a simple ESOP efficiently by substituting these MESOPs into the sub-functions. Because an arbitrary function for n variables can be decomposed into a sum of 2^{n-k} functions of k variables by the Shannon expansion, a simple ESOP can be obtained quickly by using MESOPs of k variables. As for $k \leq 4$, MESOPs of all the functions have been obtained by an exhaustive method[4]. But, in the case of $k \geq 5$, the number of the functions is too large.

Table 1: Numbers of 4-variable functions requiring t products in AND-ORs and AND-EXORs

t	AND-OR	AND-EXOR
0	1	1
1	81	81
2	1804	2268
3	13472	21744
4	28904	37530
5	17032	3888
6	3704	24
7	512	0
8	26	0
Average # of products	4.13	3.66

For obtaining MESOPs for n-variable functions, we can drastically reduce the number of functions to consider by using the LP equivalence class[1,8]. The authors have given the table of LP equivalence representative functions for five variables[5]. In this paper, we introduce the LP characteristic vector, and prove that the vector uniquely specifies the LP equivalence class of a five or fewer variable function. When we use the table of MESOPs of the LP equivalence representative functions in the minimization or simplification of ESOPs, we have to identify the equivalence class of a given function. In the case of $n \leq 4$, all the equivalent functions of a given function can be identified by a table look-up method. But, in the case of $n \geq 5$, the same approach is unrealistic because the number of the functions is too large. Also, a naive method based on the definition of the LP equivalence relation is very time consuming because the number of combinations is proportional to $n! \cdot 6^n$. But, we can do this with complexity $O(n \cdot 3^n)$ by using the LP characteristic vector. Thus, the minimization for 5-variable ESOPs can be done by the table look-up method. Also, we show that ESOPs require, on the average, fewer products than SOPs for 5-variable functions.

2 Definitions and Basic Properties of Minimum ESOPs

Definition 1 x and $\bar{x}$ are *literals* of a variable x. Let $S_i \subseteq \{0,1\}$ and $S_i \neq \emptyset$ $(i = 1, 2, \cdots, n)$. $T = x_1^{S1} x_2^{S2} \cdots x_n^{Sn}$ is a *product term*, where $x_i^{\{0\}} = \bar{x}_i$, $x_i^{\{1\}} = x_i$, $x_i^{\{0,1\}} = 1$ and $x_i^{\emptyset} = 0$. For simplicity, $x_i^{\{0\}}$, $x_i^{\{1\}}$ and $x_i^{\{0,1\}}$ are denoted by x_i^0, x_i^1 and x_i^2, respectively.

Definition 2 Product terms combined with OR operators form a *Sum-Of-Products expression(SOP)*. Product terms combined with EXOR operators form an *Exclusive-or Sum-Of-Products expression(ESOP)*. An SOP for f is said to be a *minimum SOP (or MSOP)* for f if the number of products is the minimum. An ESOP for f is said to be a *minimum ESOP (or MESOP)* if the number of products is the minimum. The number of products in an ESOP F is denoted by $\tau(F)$. The number of products in an MESOP for f is denoted by $\tau(f)$. The maximum number of the products in MESOPs for functions of n variables is denoted by $\psi(n)$.

Example 1 Let $\begin{bmatrix} f_0 \\ f_1 \end{bmatrix}$ be a vector representation of $f = \bar{x}_i \cdot f_0 \oplus x_i \cdot f_1$, and let $M = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ be a binary matrix. We perform matrix multiplication in the usual way except that $\cdot$ replaces multiplication and $\oplus$ replaces addition. That is $\begin{bmatrix} g_0 \\ g_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}$, yields $\begin{bmatrix} f_0 \\ f_0 \oplus f_1 \end{bmatrix}$, which is the vector representation of $g = \bar{x}_i \cdot g_0 \oplus x_i \cdot g_1 = \bar{x}_i \cdot f_0 \oplus x_i(f_0 \oplus f_1) = 1 \cdot f_0 \oplus x_i \cdot f_1$. The matrix multiplication has, in effect, transformed $\bar{x}_i \cdot f_0 \oplus x_i \cdot f_1$ into $1 \cdot f_0 \oplus x_i \cdot f_1$. The above transformation shows that interchanging the literals $\bar{x}_i$ and 1 in the ESOP for f gives the ESOP for g. (End of example)

Definition 3 A regular matrix is one whose determinant is nonzero. For example, $M = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ is regular.

Lemma 1 [8] Let a function f be expanded as $f = \bar{x}_i \cdot f_0 \oplus x_i \cdot f_1$. Let $\begin{bmatrix} g_0 \\ g_1 \end{bmatrix} = M \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}$, where M is a 2×2 regular matrix. Let F be an arbitrary ESOP for f, and G be the ESOP which is obtained by the transformation of literals of a variable x_i in F by M. Then, G represents $g = \bar{x}_i \cdot g_0 \oplus x_i \cdot g_1$.

Remark 1 Lemma 1 shows that the transformation of the sub-functions for f by the regular matrix is equivalent to the transformation of the literals in the ESOP for f by the same regular matrix. The number of 2×2 regular matrices is 6, and the transformations by these matrices correspond to the following six literal transformations: 1)identity, 2)$x_i \leftrightarrow \bar{x}_i$, 3)$x_i \leftrightarrow 1$, 4)$\bar{x}_i \leftrightarrow 1$, 5)$\bar{x}_i \rightarrow 1 \rightarrow x_i \rightarrow \bar{x}_i$ and 6)$\bar{x}_i \rightarrow x_i \rightarrow 1 \rightarrow \bar{x}_i$.

Theorem 1 [8] Let a function f be expanded as $f = \bar{x}_i \cdot f_0 \oplus x_i \cdot f_1$. Let $\begin{bmatrix} g_0 \\ g_1 \end{bmatrix} = M \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}$, where M is a 2×2 regular matrix. Then, $\tau(f)=\tau(g)$, where $g = \bar{x}_i \cdot g_0 \oplus x_i \cdot g_1$.

Definition 4 The relation $\sim$ satisfying the following conditions is called the *LP equivalence relation.*
1)$f \sim f$.
2)If $f_1 = f(\cdots, x_i, \cdots, x_j, \cdots)$ and $f_2 = f(\cdots, x_j, \cdots, x_i, \cdots)$, then $f_1 \sim f_2$.
3)Let a function $f(x_1, x_2, \cdots, x_n)$ be expanded as $f = \bar{x}_i \cdot f_0 \oplus x_i \cdot f_1$ by a variable x_i. Let $\begin{bmatrix} g_0 \\ g_1 \end{bmatrix} = M \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}$, where M is a 2×2 regular matrix. Then $g \sim f$, where $g = \bar{x}_i \cdot g_0 \oplus x_i \cdot g_1$.

Example 2 Consider a two-variable function $f(x,y) = x \oplus y$. Examples of the literal transformations for f are as follows: $x \oplus y \sim x \oplus \bar{y}$ (interchange of literals y and $\bar{y}$), $x \oplus \bar{y} \sim 1 \oplus x\bar{y}$ (interchange of literals x and 1). Table 2 shows the LP equivalence classes of 2-variable functions. (End of example)

Table 2: LP equivalence classes of 2-variable functions

equivalence classes	functions			
1	0			
2	$\bar{x}\bar{y}$	$\bar{x}y$	$x\bar{y}$	xy
	$\bar{x}$	x	$\bar{y}$	y
	1			
3	$x \oplus \bar{y}$	$x \oplus y$		
	$1 \oplus \bar{x}\bar{y}$	$1 \oplus \bar{x}y$	$1 \oplus x\bar{y}$	$1 \oplus xy$

Remark 2 The number of different 2×2 regular matrices is 6, and the number of n-variable functions which are *LP* equivalent to an n-variable function is at most $n!6^n$.

Table 3: Numbers of equivalence classes under different equivalence relations

n	1	2	3	4	5	6
Total # of functions	4	16	256	65536	4.3×10^9	1.8×10^{19}
P equivalence classes	4	12	80	3984	37333248	2.5×10^{16}
NP equivalence classes	3	6	22	402	1228158	4.0×10^{14}
NPN equivalence classes	2	4	14	222	616126	2.0×10^{14}
LP equivalence classes	2	3	6	30	6936	$\geq 5.5 \times 10^{11}$

Example 3 The total number of the three variable functions $f(x, y, z)$ is 256. These functions can be partitioned into 6 LP equivalence classes. The MESOPs for the representative functions are 0, $\bar{x}\bar{y}\bar{z}$, $\bar{x}y \oplus \bar{x}z$, $\bar{x} \oplus \bar{y}\bar{z} \oplus \bar{x}yz$, $x\bar{y}\bar{z} \oplus \bar{x}yz$ and $\bar{x} \oplus y\bar{z} \oplus x\bar{y}z$. MESOPs for other functions can be obtained by the permutation of variables and the literal transformations of the MESOP for the representative functions. (End of example)

Table 3 compares numbers of equivalence classes under different equivalence relations.

3 MESOPs for the LP Equivalence Representative Functions

In this section, we show an efficient algorithm to obtain the MESOPs for the representative functions.

Definition 5 The set of the representative functions for n variables is denoted by $LP(n)$. The set of the representative functions, whose number of products in the MESOP is t, is denoted by $LP(n,t)$. The set of MESOPs for functions of $LP(n,t)$ is denoted by $M(n,t)$. $M(n,t)$ can be represented by $M(n,t) = \{Fm(f)|f \in LP(n,t)\}$.

Definition 6 The function which is represented by an ESOP F is denoted by $r(F)$. An MESOP for f is denoted by $Fm(f)$. The representative function for f is denoted by $lp(f)$.

Definition 7 The set of all the products for n variables is denoted by $PT(n)$.

Lemma 2 Let F be an MESOP for a function f. Let p be one of the products in F, and G be the ESOP which is obtained by deleting p from F. Then G is an MESOP.

Proof: Let $\tau(F) = t$. Suppose that G is not an MESOP. Then, there exists G' which represents the same function as G and $\tau(G') \leq t-2$. Let $H = G' \oplus p$. Then, we have $\tau(H) \leq t-1$. This means that the function f can be represented by an ESOP whose number of products is at most $t-1$, because H represents the function f. This contradicts the hypothesis. Hence the lemma. (Q.E.D.)

Lemma 3 Let F be an MESOP whose number of products is t, and p be a product term. Let $G = F \oplus p$ and $g = r(G)$. Then, $t-1 \leq \tau(g) \leq t+1$.

Proof: 1)The upper bound exists because $\tau(F) = t$. 2)Suppose that $\tau(g) \leq t-2$. Let G' be the MESOP for G. By the hypothesis of the lemma, we have $\tau(G') \leq t-2$. Next, we consider the ESOP $H = G' \oplus p$. We have

$$\tau(H) \leq t-1. \tag{1}$$

Because both H and F represent the same function and F is the MESOP, we have

$$\tau(H) = t. \tag{2}$$

However, (1) contradicts (2). Therefore, we have $\tau(g) \geq t-1$. From 1) and 2), we have the lemma. (Q.E.D.)

Lemma 4 Let $H = \{F \oplus q | F \in M(n,t), q \in PT(n)\}$ and $R = \{lp(f)|f = r(F), F \in H\}$. Then $LP(n, t+1) \subseteq R$.

Proof: Let $g \in LP(n, t+1)$ and G be an MESOP for g. Let p be one of the products in G, and K be the ESOP which is obtained by deleting the product p from G. Then we have $G = K \oplus p$. From Lemma 2, K is an MESOP. Let K represent the function k, and α be the procedure to obtain $lp(k)$ from k. Let K' be the ESOP applying the transformation α to the literals of K, and p' be the product applying the tranformation α to the literals of p. Then, we have $K' \in M(n,t)$ and $p' \in PT(n)$. Let $G' = K' \oplus P'$. Obviously, we have $G' \in H$. Let G' represents the function g'. By applying the inverse transformation of α to g', we obtain the representative function g. Therefore, we have $g \in R$. Hence the lemma. (Q.E.D.)

3.1 Algorithm to obtain M(n,t)

In this part, we show an algorithm to obtain the MESOPs of the LP equivalence representative functions by using the properties of the MESOPs described in the previous section.

Algorithm 1 (Computation of M(n,t))
Suppose that $LP(n)$ and $M(n, 1)$ are given.

1)$t \leftarrow 1$.
2)Generate the set of ESOPs by $EXOR$ing one product and an each MESOP of $M(n,t)$.
3)Obtain the set of LP equivalence representative functions of the functions which are represented by each ESOP of 2).
4)Delete the members of the set $\{LP(n,s)|s \leq t\}$ from the set of 3).
5)The set of the representative functions which is obtained by the step 4) is $LP(n,t+1)$. We obtain $M(n,t+1)$ by using ESOPs from which the representative functions are obtained.
6)If the MESOPs of all the representative functions are obtained, the algorithm stops.
7)$t \leftarrow t+1$. Go to 2).

Remark 3 1)$M(n,t)$ denotes the set of products of n variables. The number of the different products is 3^n.
2)From Lemma 4, in order to obtain $M(n,t+1)$, it is sufficient to consider the set of ESOPs that are obtained by $EXOR$ing all the one product with each member of $M(n,t)$.
4)From Lemma 3, it is quite probable that the part of ESOPs which are obtained by the step 2) are not MESOPs. Therefore, we delete these ESOPs.

An upper bound on the number of combinations to consider for obtaining $M(n,t)$ by Algorithm 1 is $m = U_n 3^n \psi(n)$, where $U_n = |LP(n)|$. Because $U_n \approx 2^{2^n}/(n!6^n)$ and $\psi(n) \leq 2^{n-2}$ $(n \geq 6)$[5], we have $m \leq 2^{2^n-2}/n!$ $(n \geq 6)$. Because $U_n = 6936$ and $\psi(5) \leq 9$ for $n=5$, we have $m \leq 1.5 \times 10^7$. On the other hand, when the exhaustive method is applied, the number of combinations to consider is

$$s = \sum_{k=1}^{\psi(n)} {}_{3^n}C_k.$$

When $n=5$, we have $s \approx 7 \times 10^{15}$. Thus, Algorithm 1 is more efficient than exhaustive search.

4 LP Characteristic Vector and its Applications

There are about 4.3×10^9 different functions of five variables. These functions can be classified into 6936 classes using the LP equivalence relation, and we have obtained an MESOP for each equivalence class. In this section, we introduce the LP characteristic vector, which is unique to each LP equivalence class for $n \leq 5$. Therefore, the LP characteristic vector of a given function identifies the LP equivalence class. Thus, the MESOP of a given function can be found by a table look-up method.

4.1 Properties of LP characteristic vector

Theorem 2 *(Expansion theorem)* An arbitrary n-variable function $f(x_1, x_2, \cdots, x_n)$ can be expanded as

$$f = f_0 \oplus x_i \cdot f_2, \tag{3}$$

$$f = \bar{x}_i \cdot f_2 \oplus f_1, or \tag{4}$$

$$f = \bar{x}_i \cdot f_0 \oplus x_i \cdot f_1. \tag{5}$$

where $f_0 = f(x_i = 0)$, $f_1 = f(x_i = 1)$ and $f_2 = f_0 \oplus f_1$.

Proof: (5) is the Shannon expansion. By setting $\bar{x} = 1 \oplus x_i$ in (5), we have $f = (1 \oplus x_i)f_0 \oplus x_i f_1 = f_0 \oplus x_i(f_0 \oplus f_1) = f_0 \oplus x_i f_2$. Also, by setting $x_i = 1 \oplus \bar{x}_i$ in (5), we have $f = \bar{x}_i f_0 \oplus (1 \oplus \bar{x}_i) f_1 = \bar{x}_i(f_0 \oplus f_1) \oplus f_1 = \bar{x}_i f_2 \oplus f_1$. (Q.E.D.)

Definition 8 The expansions of a given function described by (3), (4) and (5) in Theorem 2 are called *Type 0 expansions*, *Type 1 expansions* and *Type 2 expansions*, respectively. The *expansion vector* $\boldsymbol{a} = (a_1, a_2, \cdots, a_n)$ of an n-variable function $f(x_1, x_2, \cdots, x_n)$ defines expansions as follows:
when $a_i = j$, we use Type j expansion, where $(i = 1, 2, \cdots, n)$, and $a_i, j \in \{0, 1, 2\}$.

By Definition 8, we have the following:

Lemma 5 When an n-variable function $f(x_1, x_2, \cdots, x_n)$ is expanded by an expansion vector $\boldsymbol{a} = (a_1, a_2, \cdots, a_n)$, the set of possible products is $\{x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n} | b_i \in \{0,1,2\}, b_i \neq a_i (i = 1, 2, \cdots, n)\}$.

Theorem 2 shows that an expansion of n-variable function on some variable produces three sub-functions. Repeating this expansion n times produces 3^n coefficients.

Definition 9 Let $f(x_1, x_2, \cdots, x_n)$ be an n-variable function. The *Extended Truth Table* for f, denoted by $ETT(f)$, has 3^n elements, and satisfies the following conditions: The indices which indicate the entries of $ETT(f)$ are represented by an n-trit vector called the *index vector*. That is expressed as $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)$, where $c_i \in \{0, 1, 2\} (i = 1, 2, \cdots, n)$. When f is expanded by the expansion vector $\boldsymbol{a} = (a_1, a_2, \cdots, a_n)$, the coefficient of the product $x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ is stored in the $\boldsymbol{c}$-th entry of $ETT(f)$, where $c_i = (a_i + b_i + 1) mod\ 3$.

Example 4 Let $f(x_1, x_2) = f_0 \bar{x}_1 \bar{x}_2 \oplus f_1 \bar{x}_1 x_2 \oplus f_2 x_1 \bar{x}_2 \oplus f_3 x_1 x_2$. Expansions of f by a vector $\boldsymbol{a}$ are as follows:
When $\boldsymbol{a} = (2,2)$, $f = f_0 \bar{x}_1 \bar{x}_2 \oplus f_1 \bar{x}_1 x_2 \oplus f_2 x_1 \bar{x}_2 \oplus f_3 x_1 x_2$.
When $\boldsymbol{a} = (2,1)$, $f = (f_0 \oplus f_1)\bar{x}_1 \bar{x}_2 \oplus (f_2 \oplus f_3) x_1 \bar{x}_2 \oplus f_1 \bar{x}_1 \oplus f_3 x_1$.
When $\boldsymbol{a} = (2,0)$, $f = (f_0 \oplus f_1)\bar{x}_1 x_2 \oplus (f_2 \oplus f_3) x_1 x_2 \oplus f_0 \bar{x}_1 \oplus f_2 x_1$.

When $\boldsymbol{a} = (1,2)$, $f = (f_0 \oplus f_2)\bar{x}_1\bar{x}_2 \oplus (f_1 \oplus f_3)\bar{x}_1 x_2 \oplus f_2\bar{x}_2 \oplus f_3 x_2$.
When $\boldsymbol{a} = (1,1)$, $f = (f_0 \oplus f_1 \oplus f_2 \oplus f_3)\bar{x}_1\bar{x}_2 \oplus (f_1 \oplus f_3)\bar{x}_1 \oplus (f_2 \oplus f_3)\bar{x}_2 \oplus f_3$.
When $\boldsymbol{a} = (1,0)$, $f = (f_0 \oplus f_1 \oplus f_2 \oplus f_3)\bar{x}_1 x_2 \oplus (f_1 \oplus f_2)\bar{x}_1 \oplus (f_2 \oplus f_3)x_2 \oplus f_2$.
When $\boldsymbol{a} = (0,2)$, $f = (f_0 \oplus f_2)x_1\bar{x}_2 \oplus (f_1 \oplus f_3)x_1 x_2 \oplus f_0\bar{x}_2 \oplus f_1 x_2$.
When $\boldsymbol{a} = (0,1)$, $f = (f_0 \oplus f_1 \oplus f_2 \oplus f_3)x_1\bar{x}_2 \oplus (f_1 \oplus f_3)x_1 \oplus (f_0 \oplus f_1)\bar{x}_2 \oplus f_1$.
When $\boldsymbol{a} = (0,0)$, $f = (f_0 \oplus f_1 \oplus f_2 \oplus f_3)x_1 x_2 \oplus (f_0 \oplus f_2)x_1 \oplus (f_0 \oplus f_1)x_2 \oplus f_0$.
$ETT(f)$ is shown in Table 4. For example, when $\boldsymbol{a} = (0,1)$, the coefficient of $x_1^1 x_2^2$, that is $(f_1 \oplus f_3)$, is stored in $ETT(f)$ with the index $\boldsymbol{c} = (2,1)$, because $(0,1) + (1,2) + (1,1) = (2,4) \equiv (2,1)(mod\ 3)$. (End of exmaple)

An algorithm to obtain the extented truth table will be shown in Algorithm 2.

Table 4: Extended truth table for 2-variable function

c	$ETT(f)$
00	f_0
01	f_1
02	$f_0 \oplus f_1$
10	f_2
11	f_3
12	$f_2 \oplus f_3$
20	$f_0 \oplus f_2$
21	$f_1 \oplus f_3$
22	$f_0 \oplus f_1 \oplus f_2 \oplus f_3$

Table 5: Extended weight table for 2-variable function

c	$EWT(f)$
00	2
01	1
02	1
10	2
11	1
12	1
20	4
21	2
22	2

Definition 10 The number of the products in the expansion of an n-variable function f by an expansion vector $\boldsymbol{a}$ is denoted by $w(f, \boldsymbol{a})$, where $0 \leq w \leq 2^n$.

Definition 11 Let $f(x_1, x_2, \cdots, x_n)$ be an n-variable function. The *Extended Weight Table* for f, denoted by $EWT(f)$, has 3^n elements, and satisfies the following conditions: The indices of $EWT(f)$ are represented by n-trit vectors $\boldsymbol{a} = (a_1, a_2, \cdots, a_n)$, where $a_i \in \{0,1,2\}(i = 1, 2, \cdots, n)$. The $\boldsymbol{a}$-th entry of $EWT(f)$ is equal to $w(f, \boldsymbol{a})$.

Example 5 Table 5 is the extended weight table for the function, where $f_0 = 1$, $f_1 = 0$, $f_2 = 1$ and $f_3 = 0$ in Example 4. For example, $(2,0)$-th entry of $EWT(f)$ is 4 because the number of products is 4 according to Example 4. (End of example)

Definition 12 Let f be an arbitrary n-variable function. The LP characteristic vector is one that is obtained from $EWT(f)$ by rearranging the elements in ascending order, and is denoted by $LPV(f)$.

An algorithm to obtain the extended weight table will be shown in Algorithm 3.

Lemma 6 Let $f(x_1, x_2, \cdots, x_n)$ be an arbitrary n-variable function. Let the ESOP, which is obtained by interchanging the literals $\bar{x}_i$ and x_i of the ESOP for f, represent a function g. Then $LPV(f) = LPV(g)$.

Proof: Note that f and g can be represented as $f = \bar{x}_i f_{i0} \oplus x_i f_{i1}$ and $g = x_i f_{i0} \oplus \bar{x}_i f_{i1} = \bar{x}_i g_{i0} \oplus x_i g_{i1}$, respectively. Therefore, we have $g_{i0} = f_{i1}$, $g_{i1} = f_{i0}$ and $g_{i2} = g_{i0} \oplus g_{i1} = f_{i1} \oplus f_{i0} = f_{i2}$. In other words, g is obtained by interchanging expansion 0 and expansion 1 in the expansions of f. By setting $\boldsymbol{a} = (a_1, a_2, \cdots, a_i, \cdots, a_n)$, $\boldsymbol{b} = (a_1, a_2, \cdots, b_i, \cdots, a_n)$ and $b_i = \begin{cases} 1 - a_i & (\text{if } a_i = 0 \text{ or } 1) \\ a_i & (\text{if } a_i = 2) \end{cases}$, we have $w(f, \boldsymbol{a}) = w(g, \boldsymbol{b})$. That is, $EWT(g)$ can be obtained by interchanging the entries of $EWT(f)$. Hence the lemma. (Q.E.D.)

Lemma 7 Let $f(x_1, x_2, \cdots, x_n)$ be an arbitrary n-variable function. Let the ESOP, which is obtained by interchanging the literals $\bar{x}_i$ and 1 in an ESOP for f, represent a function g. Then, $LPV(f) = LPV(g)$.

Proof: Note that f and g can be represented as $f = \bar{x}_i f_{i0} \oplus x_i f_{i1}$ and $g = 1 \cdot f_{i0} \oplus x_i f_{i1} = \bar{x}_i f_{i0} \oplus x_i(f_{i0} \oplus f_{i1}) = \bar{x}_i g_{i0} \oplus x_i g_{i1}$, respectively. Therefore, we have $g_{i0} = f_{i0}$, $g_{i1} = f_{i0} \oplus f_{i1} = f_{i2}$ and $g_{i2} = g_{i0} \oplus g_{i1} = f_{i0} \oplus (f_{i0} \oplus f_{i1}) = f_{i1}$. In other words, g is obtained by interchanging expansion 0 and expansion 2 in the expansions of f. By setting $\boldsymbol{a} = (a_1, a_2, \cdots, a_i, \cdots, a_n)$, $\boldsymbol{b} = (a_1, a_2, \cdots, b_i, \cdots, a_n)$ and $b_i = \begin{cases} 2 - a_i & (\text{if } a_i = 0 \text{ or } 2) \\ a_i & (\text{if } a_i = 1) \end{cases}$, we have $w(f, \boldsymbol{a}) = w(g, \boldsymbol{b})$. That is, $EWT(g)$ is obtained by interchanging the entries of $EWT(f)$. Hence the lemma. (Q.E.D.)

Lemma 8 Let $f(x_1, x_2, \cdots, x_n)$ be an arbitrary n-variable function. Let the ESOP, which is obtained by interchanging the literals x_i and 1 in an ESOP for f, represent a function g. Then, $LPV(f) = LPV(g)$.

Proof: Similar to Lemma 7. (Q.E.D.)

Lemma 9 Let $f(x_1, x_2, \cdots, x_n)$ be an arbitrary n-variable function. Let g be a function which is obtained by interchanging the variables x_i and x_j in f. Then, $LPV(f) = LPV(g)$.

Proof: Let F be an ESOP which is obtained by expanding f by an expansion vector $\boldsymbol{a} = (a_1, a_2, \cdots, a_n)$, and G be an ESOP which is obtained by interchanging the literals x_i and x_j in F. Note that G represents g. Then, we have $\tau(F) = \tau(G)$. By setting $\boldsymbol{a} = (\cdots, a_i, \cdots, a_j, \cdots)$ and $\boldsymbol{b} = (\cdots, a_j, \cdots, a_i, \cdots)$, we have

$$w(g, \boldsymbol{a}) = w(f, \boldsymbol{b}) \quad (6)$$

Because (6) holds for all possible $\boldsymbol{a}$, we have $EWT(g)_{\boldsymbol{a}} = EWT(f)_{\boldsymbol{b}}$. Hence the lemma. (Q.E.D.)

Lemma 10 $f \sim g \rightarrow LPV(f) = LPV(g)$.

Proof: From Lemmas 6 to 9, the LP characteristic vector does not change by the interchange of literals and the permutation of variables. Hence the lemma. (Q.E.D.)

Lemma 11 $LPV(f) = LPV(g) \rightarrow f \sim g \ (n \leq 5)$.

Proof: We prove $f \not\sim g \rightarrow LPV(f) \neq LPV(g)$. All the LPVs of the LP equivalence representative functions for five or fewer variables are computed, and are found to be distinct. Hence the lemma. (Q.E.D.)

Theorem 3 For five or fewer variable functions f and g, $f \sim g \leftrightarrow LPV(f) = LPV(g)$.

Proof: From Lemmas 10 and 11, we have the theorem. (Q.E.D.)

From Theorem 3, when $n \leq 5$, the LP equivalence class of a given function is identified by the LP characteristic vector. Table 6 shows extended weight tables for $\bar{x}\bar{y}$, xy, $\bar{x}$, y and 1. When the elements of the vectors are rearranged in ascending order, then all the vectors become (1,1,1,1,2,2,2,2,4). This is the LP characteristic vector for these functions.

4.2 Algorithm for LP characteristic vector

The procedure for obtaining the LP characteristic vector of a given function is as follows:
1)Obtain the extended truth table from the truth table.
2)Obtain the extended weight table from the extended truth table.
3)Sort the extended weight table in ascending order. This is the LP characteristic vector.

Table 6: Extended weight tables of 2-variable functions

function	Extended weight tables
$\bar{x}\bar{y}$	4 2 2 2 1 1 2 1 1
xy	1 2 1 2 4 2 1 2 1
$\bar{x}$	2 2 4 1 1 2 1 1 2
y	1 2 1 1 2 1 2 4 2
1	1 1 2 1 1 2 2 2 4

Let f be a truth table of a n variable function, E_f be an extended truth table and W_f be an extended weight table. The indices of f are represented by n-bit vectors. The indices of E_f and W_f are represented by n-trit vectors.

Algorithm 2 (Extended Truth Table)
1)For the entries with index vectors consisting of either 0 or 1:
$E_f(a_1, a_2, \cdots, a_n) \leftarrow f(a_1, a_2, \cdots, a_n)$,
where $a_k \in \{0, 1\} (k = 1, 2, \cdots, n)$.
2)For the entries with the index vectors where only one digit, say i-th, is 2:
$E_f(a_1, \cdots, 2, \cdots a_n) \leftarrow$
$E_f(a_1, \cdots, 0, \cdots, a_n) \oplus E_f(a_1, \cdots, 1, \cdots, a_n)$
where $a_k = \{0, 1\} (k = 1, 2, \cdots, n, k \neq i)$ and $\oplus$ denotes bitwise EXOR operation.
3)For the entries with the index vector where two digits, say i-th and j-th, are 2:

$$E_f(a_1, \cdots, \underbrace{2}_{i}, \cdots, \underbrace{2}_{j}, \cdots, a_n)$$
$$\leftarrow E_f(a_1, \cdots, \underbrace{0}_{i}, \cdots, \underbrace{2}_{j}, \cdots, a_n)$$
$$\oplus\ E_f(a_1, \cdots, \underbrace{1}_{i}, \cdots, \underbrace{2}_{j}, \cdots, a_n),$$

where $a_k = \{0, 1\} (k = 1, 2, \cdots, n, k \neq i, k \neq j)$.
4)For the entries with index vectors where three or more digits are 2:
Obtain similar to the above.

Algorithm 3 (Extended Weight Table)

for $i = 0$ **to** 3^n **do**
  {**if**($E_f(i) = 0$) **then** $W_f(i) \leftarrow 0$
    else $W_f(i) \leftarrow 1$}
for $k = 0$ **to** n **do** {
  for $j = 0$ **to** $3^{k-1} - 1$ **do** {
    for $i = 0$ **to** $3^{n-k} - 1$ **do** {
      $i_0 \leftarrow 3^{n-k}(3j + 0) + i$;
      $i_1 \leftarrow 3^{n-k}(3j + 1) + i$;
      $i_2 \leftarrow 3^{n-k}(3j + 2) + i$;
      $W_1(i_0) \leftarrow W_f(i_0) + W_f(i_2)$;

$$
\begin{array}{l}
\qquad\qquad\qquad W_1(i_1) \leftarrow W_f(i_1) + W_f(i_2); \\
\qquad\qquad\qquad W_1(i_2) \leftarrow W_f(i_0) + W_f(i_1); \\
\qquad\qquad \} \\
\qquad\qquad \textbf{for } i = 0 \textbf{ to } 3^n \textbf{ do} \\
\qquad\qquad\qquad \{W_f(i) \leftarrow W_1(i)\} \\
\qquad \} \\
\}
\end{array}
$$

The complexities of the algorithms are as follows.

Lemma 12 The complexities for obtaining the extended truth table and the extended weight table are $O(3^n)$, and $O(n3^n)$, respectively.

Theorem 4 The complexity for obtaining the LP characteristic vector is $O(n3^n)$.

Proof: The LP characteristic vector can be obtained by sorting the extended weight table in ascending order. Hence the theorem. (Q.E.D.)

Remark 4 To decide the equivalence class of a given function, the computation time is proportional to $n! \cdot 6^n$ if we use a naive algorithm based on the definition of the LP equivalence relation, while only $n3^n (n \leq 5)$ if we use the LP characteristic vector.

4.3 Algorithm to obtain the MESOP

A MESOP of the given function for 5 variables is obtained by the following:

Algorithm 4 (MESOP)
1)Obtain the LP characteristic vector $LPV(f)$ of a given function f.
2)Decide the LP equivalence class of f by using $LPV(f)$.
3)Obtain $\tau(f)$ by using the table of MESOPs for the representative functions[5].
4)Simplify f by $EXMIN2$[10]. Let F be the simplified ESOP.
5)If $\tau(f) = \tau(F)$, then F is the minimum, else obtain the MESOP of the given function by the inverse transformation of the MESOP of the representative function.

Remark 5 1)By Theorem 3, there is a one-to-one correspondence between an LP characteristic vector and an LP equivalence class for a function of five or fewer variables.
3)Obtain the number of products in an MESOP for a given function.
5)$EXMIN2$ is a heuristic simplification program for ESOPs. It can simplify ESOPs quickly, but does not always produce minimum solutions. If the number of products in the simplified ESOP by $EXMIN2$ is found to be minimum, then the MESOP is obtained. Else, the simplified result is not the minimum. Then, the MESOP for a given function is obtained by the inverse transformation of the MESOP for the representative function. In this case, the transformation to obtain the LP equivalence representative function from a given function is needed. To do this, it takes computation time proportional to $n! \cdot 6^n$ by the algorithm based on the LP equivalence relation. For five variable functions, this takes about 7 seconds by an HP9000/720 workstation.

5 Experimental Results and Discussions

We have obtained the table of MESOPs for the LP equivalence representative functions of five variables[5]. ESOPs for randomly generated functions of five variables are minimized by using Algorithm 4. When $EXMIN2$ produces the minimum solution, the computing time is about 90 milliseconds per function. When $EXMIN2$ produces non-minimum and the MESOP is obtained by the inverse transformation of the MESOP for the representative function, the computing time is about 7 seconds per function by an HP9000/720 workstation. By using the LP characteristic vector, the equivalence class of a 5-variable function is obtained very quickly. Because the number of the products in the MESOP for the representative function is obtained by a table look-up method, we can decide whether $EXMIN2$ produced the minimum or not very quickly. If the number of the products in the simplified ESOP is equal to that of the MESOP which is obtained by the table look-up method, then the simplified ESOP is the minimum. In this case, we have obtained the MESOP. On the other hand, if not, we have to obtain the MESOP by the inverse transformation. Fig. 1 shows the average computing time for the MESOP of an arbitrary 5-variable function.

It has long been conjectured that ESOPs require, on the average, fewer products than SOPs to realize logic functions. To confirm this conjecture for five variable functions, we obtained the average number of products in MESOPs and MSOPs for randomly generated functions. The set of five variable functions is partitioned into classes according to the number of minterms. For each class, we generated 1000 random functions. Fig. 2 shows the average number of products to realize these functions in MESOPs and MSOPs. MSOPs were obtained by Quine-McCluskey method. Fig. 2 shows that, on the average, MESOPs require fewer products than MSOPs to realize five variable functions. Also, we proved that MESOPs require only 9 products to realize an arbitrary function of five variables[5], while MSOPs require 16 products. From these facts, we confirmed the above conjecture for the case of 5-variable functions.

6 Conclusion

In this paper, we introduced the LP characteristic vector of a logic function and showed that the vector

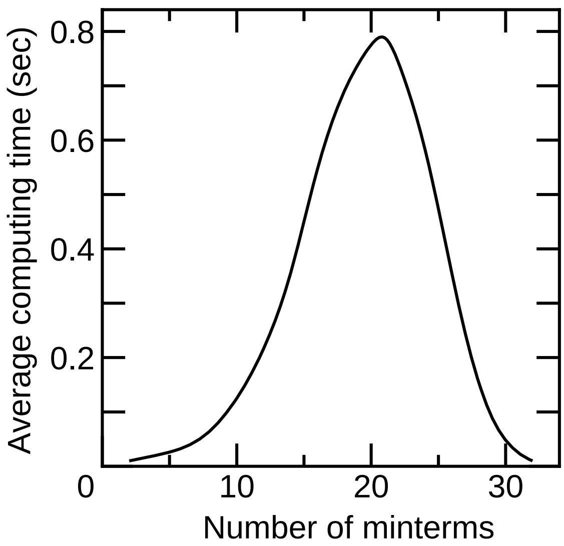

Figure 1: Average computing time for 5-variable MESOPs.

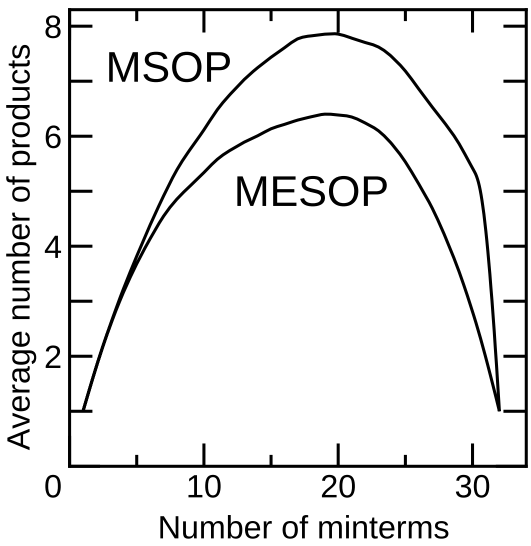

Figure 2: Average number of products in MSOPs and MESOPs of five variable functions.

uniquely specifies the equivalence class of a given n-variable function, when $n \leq 5$. Also, we presented an efficient algorithm to find an MESOP for a 5-variable function by using the LP characteristic vector. For the case of 6 or more variable functions, we can decompose a given function into sums of 2^{n-5} sub-functions by using Shannon expansions. By substituting MESOPs for five variables into these sub-functions, we can obtain a simple ESOP quickly. The LP characteristic vectors of functions for up to 14 variables can be obtained easily by an ordinary computer, so other applications can be expected.

Acknowledgements

This work was supported in part by Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan. The authors also thank Prof. Jon T. Butler whose comments made considerable improvement in the presentation of the paper.

References

[1] M. Davio, J.P. Deschamps and A. Thayse: *Discrete and Switching Functions*, McGraw-Hill International, 1978.

[2] M.A. Harrison: *Introduction to Switching and Automata Theory*, McGraw Hill, 1965.

[3] M. Helliwell and M. Perkowski: "A fast algorithm to minimize multi-output mixed-polarity generalized Reed-Muller forms", *Proc. of the 25th Design Automation Conference*, pp.427-432, June 1988.

[4] N. Koda and T. Sasao: "Four-variable AND-EXOR minimum expressions and their properties" (in Japanese), *Trans. IEICE*, Vol.J74-D-I,11, pp.765-773, Nov. 1991.(English translation: *Computers and Communications in Japan*, Vol.23, No.10, pp.27-41, May 1993)

[5] N. Koda and T. Sasao: "An upper bound on the number of product terms in AND-EXOR minimum expressions" (in Japanese), *Trans. IEICE*, Vol.J75-D-I, No.3, pp.135-142, Mar. 1992.

[6] M. Perkowski and M. Chrzanowska-Jeske: "An exact algorithm to minimize mixed-radix exclusive sums of products for incompletely specified Boolean functions", *Proc. International Sympo. on Circuits and Systems*, pp.1652-1655, May 1990.

[7] T. Sasao and P.W. Besslich: "On the complexity of MOD-2 SUM PLA's", *IEEE Trans. on Compt.*, vol.39, No.2, pp.262-266, Feb. 1990.

[8] T. Sasao: "A transformation of multiple-valued input two-valued output functions and its application to simplification of exclusive-or sum-of-products expressions", *Proc. of the 21th International Symposium on Multiple-Valued Logic*, pp.270-279, May 1991.

[9] T. Sasao: "AND-EXOR expressions and their optimization", in Sasao(ed.) *Logic Synthesis and Optimization*, Kluwer Academic Publishers, Jan. 1993.

[10] T. Sasao: "EXMIN2: A simplification algorithm for Exclusive-Or-Sum of products expressions for multiple-valued input two-valued output functions", *IEEE Trans. on CAD* (to be published).