数理解析研究所講究録
796巻 1992年 67-83

種々のＡＮＤ－ＥＸＯＲ論理式の複雑度について

笹尾 勤
＊ 九州工業大学情報工学部　電子情報工学科

あらまし　７つのＡＮＤ－ＥＸＯＲ形論理式（正極性ＲＭＥ，固定極性ＲＭＥ，クロネッカー式，擬似ＲＭＥ，擬似クロネッカー式，一般化ＲＭＥ，ＥＳＯＰ）を定義し，それぞれの論理式の関係と複雑度を明らかにする．いくつかの関数のクラスを実現するために必要な積項数について解析している．特に関数 $x_1y_1 \vee x_2y_2 \vee \cdots \vee x_ny_n$ を実現するために必要十分な積項数が 2^n-1 であることを証明している．上記の論理式の積項数を最小化するプログラムを開発し，種々の算術演算回路，乱数関数，およびすべての４および５変数関数について積項数を調べた結果を表にしている．

On the Complexity of Some Classes of AND-EXOR Expressions

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

Abstract　This paper presents 7 classes of AND-EXOR expressions:positive polarity Reed-Muller expressions, fixed polarity Reed-Muller expressions, Kronecker expressions, pseudo Reed-Muller expressions, pseudo Kronecker expressions, generalized Reed-Muller expressions and exclusive-or sum-of-products expressions(ESOPs). Relations between these classes are shown. The number of products to realize several classes of functions are analyzed. Especially, the number of products in a minimal ESOP for $x_1y_1 \vee x_2y_2 \vee \cdots \vee x_ny_n$ is shown to be 2^n-1. Optimization programs for these expressions were developed, and statistical results for arithmetic functions, randomly generated functions, and all the functions of 4 and 5 variables were obtained.

1. Introduction

It has long been conjectured that exclusive sum-of-products expressions (ESOPs) require fewer products than sum-of-products expressions (SOPs). For example, an ESOP requires only n products to represent a parity function of n variables while the SOP requires 2^{n-1}. Also, experiments using randomly generated functions show that ESOPs require, on the average, fewer products than SOPs. However, this is not always the case. There is a 2n variable function which requires $2^n -1$ products in an ESOP while only n products in an SOP.

This paper presents 7 classes of AND-EXOR expressions: positive polarity Reed-Muller expressions, fixed polarity Reed-Muller expressions, Kronecker expressions, pseudo Reed-Muller expressions, pseudo Kronecker expressions, generalized Reed-Muller expressions and exclusive-or sum-of-products expressions (ESOPs). Relations of these classes are shown. The number of products to realize several classes of functions are analyzed. Optimization programs for these expressions were developed, and statistical results for arithmetic functions, randomly generated functions, and all the functions of 4 and 5 variables were obtained. Also, we will prove that the ESOP for $x_1y_1 \vee x_2y_2 \vee \cdots \vee x_ny_n$ requires 2^n-1 products.

2. Several Classes of AND-EXOR Expressions

Many researchers defined various classes of AND-EXOR expressions, but the terminology is not unified. In this section, we define several classes and show the relations among them. Also, we propose a new class of AND-EXOR expression.

Theorem 2.1:(Expansion Theorem) An arbitrary logic function f can be expanded as either

$f= f_0 \oplus xf_2$ ---(1) , or

$f=\bar{x}f_2 \oplus f_1$ ---(2) , or

$f=\bar{x}f_0 \oplus xf_1$ ---(3), where

$f_0=f(0,x_2,\cdots,x_n)$, $f_1=f(1,x_2,\cdots,x_n)$, and $f_2=f_0 \oplus f_1$.

In the case of SOPs we can use only the type (3) expansion, which is often called a Shannon expansion. However, in the case of AND-EXOR expressions, we may use any of the three expansions. Thus, various classes of expressions exist as follows:

2.1 Positive Polarity Reed-Muller Expression (PPRME)

When we apply the type (1) expansion to all the variables, we have an expression consisting of positive literals only:

$$a_0 \oplus a_1x_1 \oplus \cdots \oplus a_nx_n \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus \cdots \oplus a_{n\,n-1}x_nx_{n-1} \oplus \cdots\cdots \oplus a_{12\cdots n}x_1x_2\cdots x_n \quad ----(4)$$

This is called a Positive Polarity Reed-Muller Expression (PPRME). Because PPRME is unique for a given function, no minimization problem exists. The average number of product terms in the PPRMEs for the n-variable functions is 2^{n-1} [SAS 90a].

2.2 Fixed Polarity Reed-Muller Expression (FPRME)

When we apply either the type (1) or the type (2) expansion to each variable, we obtain an expression similar to (4), except that either a true or a complemented literal is used for each variable. This expression is called a Fixed Polarity Reed-Muller expression (FPRME).

There are at most 2^n different FPRMEs for an n-variable function. The minimization problem is to find an expression with the minimum numbers of products among the 2^n different FPRMEs. As for minimization, two different methods are known: One requires the space and the computation time of $O(2^n)$ and $O(4^n)$, respectively [LUI 90], and the other requires the space and the computation time of $O(3^n)$ [DAV78].

2.3 Kronecker Expression (KRO)

When we apply either the type (1), (2) or (3) expansion to each variable, we obtain an expression which is more general than FPRME. This is called a Kronecker expression (KRO)[DAV78]. There are at most 3^n different KROs for an n-variable function. As an algorithm to find a KRO with the minimum number of products, a method using an extended truth table of 3^n entries and extended weight vector is known. The time and space complexity of the algorithm are $O(n \cdot 3^n)$ and $O(3^n)$, respectively [DAV 78], [LUI 90].

2.4 Pseudo Reed-Muller Expression (PSDRME)

When we apply either the type (1) or the type (2) expansion to f, we have two sub-functions. For each sub-function, we can apply either type (1) or (2) expansion. However, assume that we use different expansions for each sub-function. In this case, we have a more general expansion than a FPRME. This is called a Pseudo Reed-Muller Expression (PSDRME). In PSDRME, both true and complemented literals can appear for the same variable. There are at most 2^{2^n-1} different PSDRMEs. A minimum PSDRME can be obtained from the extended truth table. However the number of products in the expression depends on the order of the variables. This class of expressions has not be studied according to the author's knowledge.

2.5 Pseudo Kronecker Expression (PSDKRO)

When we apply either the type (1), (2) or (3) expansion to f, we have two sub-functions. For each sub-function, we can apply either the type (1), (2) or (3) expansion, and assume that we use different expansions for each sub-function. In this case, we have a more general expansion than a KRO. This is called a Pseudo Kronecker Expression (PSDKRO) [DAV78]. In PSDKRO, both true and complemented literals can appear for the same variable. There are at most 3^{2^n-1} different PSDKROs. A minimum PSDKRO can be obtained from an extended truth table. The number of products in the expression depends on the order of the variables.

2.6 Generalized Reed-Muller Expression (GRME)

In the expression of the type (4), if we can freely choose the polarities of the literals, then we have a more general expression than a FPRME. This is called a Generalized Reed-Muller Expression (GRME) [DAV78]. (Also called an inconsistent canonical form [COH62] or a canonical restricted mixed polarity form [CSA 91]).

There are at most $2^{n2^{n-1}}$ different GRMEs. No efficient minimization method is known. Note that some researchers use the term GRMEs to mean a different class of AND-EXOR expressions.

2.7 Exclusive-or Sum-of-Products Expression (ESOP)

Arbitrary product terms combined by EXORs are called an Exclusive-or Sum-of-Products Expression (ESOP). The ESOP is the most general AND-EXOR expression. There are at most 3^{tn} different ESOPs, where t is the number of the products. No efficient minimization method is known, and iterative improvement methods are used to obtain near minimal solutions. An exact minimization method was developed, but it is very time- and memory-consuming [PER90].

2.8 Relations among the classes

Theorem 2.2: Suppose that PPRME, FPRME, PSDRME, KRO, PSDKRO, GRME, and ESOP denote the set of expressions. Then the following relations hold:

① PPRME $\subset$ FPRME ② FPRME $\subset$ PSDRME ③ FPRME $\subset$ KRO
④ KRO $\subset$ PSDKRO ⑤ PSDRME $\subset$ PSDKRO ⑥ PSDRME $\subset$ GRME

(Proof) As for ①～⑤, they are trivial and follows from the definitions. As for ⑥, consider a PSDRME. It is also a GRME, and hence the proof is completed. (Q.E.D.)

Example 2.1:

○ $xy \oplus yz \oplus zx$ is a PPRME.
(all the literals are positive).

① $x\bar{y} \oplus \bar{y}z \oplus zx$ is a FPRME, but not a PPRME.
(x and z have positive literals, but y has negative literals).

② $xy \oplus \bar{y}z \oplus \bar{z}x$ is a PSDRME, but not a FPRME.
(y and z have literals of both polarities).

③ $xyz \oplus \bar{x} \cdot \bar{y} \cdot \bar{z}$ is a KRO, but not a FPRME.
(x, y and z have literals of both polarities).

④ $\bar{x} \oplus xy \oplus x\bar{y}$ is a PSDKRO, but not a KRO.

⑤ $\bar{x} \oplus xy \oplus x\bar{y}$ is a PSDKRO, but not a PSDRME.
(it contains two products of the highest degree).

⑥ $x \oplus y \oplus \bar{x} \cdot \bar{y}$ is a GRME, but not a PSDRME.

⑦ $x \oplus y \oplus \bar{x} \cdot \bar{y}$ is a GRME, but not a PSDKRO.

⑧ $xyz \oplus \bar{x} \cdot \bar{y} \cdot \bar{z}$ is a KRO, but not a GRME.
(it contains two products of the highest degree).

⑨ $x \oplus y \oplus xy \oplus \bar{x} \cdot \bar{y}$ is an ESOP, but neither GRME nor PSDKRO.

Fig. 2.1 summarizes the Theorem 2.2 and Example 2.1.

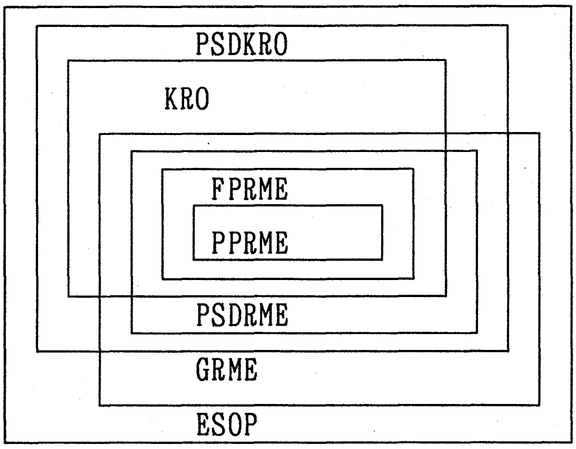

Fig 2.1 Various classes of AND-EXOR Expressions and their relation.

2.9 Complexities of the expressions for some functions

The numbers of the products to represent the function $x_1 \oplus x_2 \oplus \cdots \oplus x_n$ are n for ESOPs, and 2^{n-1} for SOPs. In the case of $\bar{x}_1\bar{x}_2\cdots\bar{x}_n$, 2^n for PPRME, and only 1 for other classes of AND-EXOR expressions [SAS90a]. In the case of $x_1x_2\cdots x_n \vee \bar{x}_1\bar{x}_2\cdots\bar{x}_n$ $(n=2r)$, 2^n-1 for PPRME, $2(2^r-1)$ for FPRME, and n for PSDRME. But for other classes only two products are sufficient [SAS90a]. The sufficient numbers of products to represent $x_1x_2 \vee x_3x_4 \vee \cdots \vee x_{n-1}x_n$ $(n=2r)$ are r for SOPs and 2^r-1 for ESOPs. The sufficient numbers of the products to represent an n-bit adder are 2^n-1 for ESOPs, $2^{n+1}+n-2$ for other classes of AND-EXOR expressions, and $6\cdot 2^n-4n-5$ for SOPs.

Table 2.1. Number of products to represent various functions (n=2r).

function	PSD KRO	PSD RME	PP RME	KRO	FP RME	ESOP	SOP
$x1 \oplus x2 \oplus \cdots \oplus xn$	n	n	n	n	n	n	2^{n-1}
$\bar{x}1\bar{x}2\cdots\bar{x}n$	1	1	2^n	1	1	1	1
$x1x2\cdots xn \vee \bar{x}1\bar{x}2\cdots\bar{x}n$	2	n	2^n-1	2	$2^{r+1}-2$	2	2
$x1x2 \vee \cdots \vee xn-1xn$	2^r-1	2^r-1	2^r-1	2^r-1	2^r-1	2^r-1	n
n bit adder	$2^{n+1}+n-2$					2^n-1	*

* $6\cdot 2^n-4n-5$

3. Complexity of ESOPs

Experimental results show that ESOPs require fewer products than SOPs to represent symmetric functions and randomly generated functions [SAS 90a]. Also, an ESOP requires only n products to represent a parity function of n variables while the SOP requires 2^{n-1} products. However, this is not always the case. There is a 2n variable function whose ESOP requires 2^n-1 products while the SOP requires n products.

In this section, we derive upper and lower bounds on the number of products in ESOPs. Especially, we show that $x_1y_1 \vee x_2y_2 \vee \cdots \vee x_ny_n$ requires 2^n-1 products. The method to prove the lower bound here use special properties which cannot be found in SOPs. Recently, an exponential lower bound of ESOP for a clique function has been obtained by using Razborov's approximation method [DAM 90] [RAZ 88].

Definition 3.1: x and $\bar{x}$ are literals of a variable x. A logical product which contains at most one literal for each variable is called a product term. 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).

Definition 3.2: A minterm is a logical product containing a literal for each variable. A minterm implying a function f is called a minterm of f.

Definition 3.3: An SOP for f is said to be a minimum SOP (or MSOP) for f if the number of the products is the minimum. An ESOP for f is said to be a minimum ESOP (or MESOP) for f if its number of products is the minimum.

Definition 3.4: The number of products in an SOP F is denoted by t(F). The number of products in an MSOP for f is denoted by t(f). 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 $\tau(f)$.

Lemma 3.1: If $f=g\oplus h$, then $\tau(f)\leqq\tau(g)+\tau(h)$.

Theorem 3.1: Let $F=\bar{x}F0\oplus xF1\oplus F2$ be an ESOP for a function f. Consider the ESOPs $G=F0\oplus xF1\oplus\bar{x}F2$ and $H=\bar{x}F0\oplus F1\oplus xF2$ which are obtained by interchanging the literals $1\rightleftarrows\bar{x}$ or $1\rightleftarrows x$ in F, respectively. Let g and h be the functions represented by G and H, respectively.
Then $\tau(f)=\tau(g)=\tau(h)$.

(Proof) Let the MESOPs for f and g be

$F^m=\bar{x}F_0^m\oplus xF_1^m\oplus F_2^m$ and $G^m=\bar{x}G_0^m\oplus xG_1^m\oplus G_2^m$, respectively.

Because F and F^m represent the same function,

$F_0\oplus F_2=F_0^m\oplus F_2^m$, and $F_1\oplus F_2=F_1^m\oplus F_2^m$.

Because G and G^m represent the same function,

$F_2\oplus F_0=G_0^m\oplus G_2^m$, $F_1\oplus F_0=G_1^m\oplus G_2^m$ and $F_1\oplus F_2=G_1^m\oplus G_0^m$.

Therefore, $F_0^m\oplus F_2^m=G_0^m\oplus G_2^m$, and $F_1^m\oplus F_2^m=G_1^m\oplus G_0^m$. Note that

$$F^m=\bar{x}F_0^m\oplus xF_1^m\oplus F_2^m=\bar{x}(G_0^m\oplus G_2^m\oplus F_2^m)\oplus x(G_1^m\oplus G_0^m\oplus F_2^m)\oplus F_2^m=\bar{x}G_2^m\oplus xG_1^m\oplus G_0^m\ .$$

From this, we have $\tau(f) \leqq \tau(g)$.
On the other hand, in a similar way, we obtain

$$G^m = \bar{x}G_0^m \oplus xG_1^m \oplus G_2^m = xF_1^m \oplus \bar{x}F_2^m \oplus F_0^m \quad .$$

So $\tau(g) \leqq \tau(f)$. Thus $\tau(g) = \tau(f)$. $\tau(f) = \tau(h)$ can be proved in a similar way. (Q. E. D.)

<u>Lemma 3.2:</u> $\tau(f_n) = \tau(g_n)$, where
$g_n = (1 \oplus x_1y_1) \cdot (1 \oplus x_2y_2) \cdots (1 \oplus x_ny_n)$, and
$f_n = (x_1 \oplus y_1) \cdot (x_2 \oplus y_2) \cdot \cdots \cdot (x_n \oplus y_n)$.
(Proof) Replace the literals as $x_i \rightleftarrows 1$ in g_n, and we have f_n.
By Theorem 3.1, $\tau(f_n) = \tau(g_n)$. (Q. E. D.)

<u>Lemma 3.3:</u> If $h_n = x_1y_1 \vee x_2y_2 \vee \cdots \vee x_ny_n$, then $h_n = g_n \oplus 1$.

(Proof) $g_n = (1 \oplus x_1y_1) \cdot (1 \oplus x_2y_2) \cdots (1 \oplus x_ny_n)$

$= \overline{(x_1y_1)} \cdot \overline{(x_2y_2)} \cdot \cdots \cdot \overline{(x_ny_n)}$.

$g_n \oplus 1 = \bar{g}_n = x_1y_1 \vee x_2y_2 \vee \cdots \vee x_ny_n = h_n$. (Q. E. D.)

<u>Lemma 3.4:</u> $\tau(h_n) \leqq 2^n - 1$.
(Proof) By Lemma 3.3, h_n can be represented as :
$h_n = (1 \oplus x_1y_1) \cdot (1 \oplus x_2y_2) \cdots (1 \oplus x_ny_n) \oplus 1$.

Using the distributive law, we have the ESOP for h_n with $2^n - 1$ products.
(Q. E. D.)

<u>Lemma 3.5:</u> $\tau(f) \geqq \tau(0{:}1)$, where $\tau(0{:}1) = \tau(f(0) \oplus f(1))$
$f(a) = f(a, x_2, x_3, \cdots, x_n)$, and $a \in \{0, 1\}$. (Proof is in Appendix)

<u>Lemma 3.6:</u> $\tau(f) \geqq \{\tau(0,0{:}0,1) + \tau(0,0{:}1,0) + \tau(1,1{:}0,1) + \tau(1,1{:}1,0)\}/2$,
where $\tau(a,b{:}c,d) = \tau(f(a,b) \oplus f(c,d))$, $f(a,b) = f(a,b,x_3,x_4,\cdots,x_n)$,
and $a,b,c,d \in \{0,1\}$. (Proof is in Appendix)

<u>Lemma 3.7:</u> $\tau(f) \geqq$
$\{\tau(0,0,0{:}0,0,1) + \tau(0,0,0{:}0,1,0) + \tau(0,0,0{:}1,0,0) + \tau(0,1,1{:}0,0,1)$
$+ \tau(0,1,1{:}0,1,0) + \tau(0,1,1{:}1,1,1) + \tau(1,0,1{:}0,0,1) + \tau(1,0,1{:}1,0,0)$
$+ \tau(1,0,1{:}1,1,1) + \tau(1,1,0{:}0,1,0) + \tau(1,1,0{:}1,0,0) + \tau(1,1,0{:}1,1,1)\}/4$,
where $f(a,b,c) = f(a,b,c,x_4,x_5,\cdots,x_n)$, $a,b,c,d,e,h \in \{0,1\}$,
and $\tau(a,b,c{:}d,e,h) = \tau(f(a,b,c) \oplus f(d,e,h))$. (Proof is in Appendix)

<u>Theorem 3.2:</u> $\tau(f_n) = 2^n$, where $f_n = (x_1 \oplus y_1) \cdot (x_2 \oplus y_2) \cdot \cdots \cdot (x_n \oplus y_n)$.
(Proof) Let f_n be f in Lemma 3.6. Because
$\tau(0,0{:}0,1) + \tau(0,0{:}1,0) + \tau(1,1{:}0,1) + \tau(1,1{:}1,0) = 4 \cdot \tau(f_{n-1})$,
we have $\tau(f_n) \geqq 2 \cdot \tau(f_{n-1})$. It is easy to see that $\tau(f_1) = 2$.
Thus $\tau(f_n) \geqq 2^n$. On the other hand, by the distributive law, we have
$\tau(f_n) \leqq 2^n$. Hence the theorem. (Q. E. D.)

<u>Lemma 3.8:</u> $|\tau(h) - \tau(\bar{h})| \leqq 1$.

(Proof) Suppose that $g=\overline{h}$. Because $g=h\oplus 1$ and $h=g\oplus 1$, we have $\tau(g)\leqq\tau(h)+1$ and $\tau(h)\leqq\tau(g)+1$ by Lemma 3.1. Therefore, we have $\tau(g)-\tau(h)\leqq 1$ and $\tau(h)-\tau(g)\leqq 1$. Hence the Lemma. (Q.E.D.)

Theorem 3.3: $\tau(h_n)=2^n-1$, where $h_n=x_1y_1 \vee x_2y_2 \vee \cdots \vee x_ny_n$.

(Proof) By Lemma 3.4, $\tau(h_n)\leqq 2^n-1$. By Theorem 3.2 and Lemma 3.2, $\tau(g_n)=2^n$. By Lemma 3.8, $|\tau(h_n)-\tau(g_n)|\leqq 1$. Thus, $\tau(g_n)-\tau(h_n)\leqq 1$ and $\tau(h_n)\geqq\tau(g_n)-1=2^n-1$. (Q.E.D.)

Theorem 3.4: $\tau(r_n)=3^n$, where $r_n=(x_1\oplus y_1\oplus z_1)\cdot(x_2\oplus y_2\oplus z_2)\cdot\cdots\cdot(x_n\oplus y_n\oplus z_n)$.

(Proof) Let f be r_n in Lemma 3.7. Note that

$$\tau(0,0,0{:}0,0,1)+\tau(0,0,0{:}0,1,0)+\tau(0,0,0{:}1,0,0)+\tau(0,1,1{:}0,0,1)+\tau(0,1,1{:}0,1,0)+\tau(0,1,1{:}1,1,1)+\tau(1,0,1{:}0,0,1)+\tau(1,0,1{:}1,0,0)+\tau(1,0,1{:}1,1,1)+\tau(1,1,0{:}0,1,0)+\tau(1,1,0{:}1,0,0)+\tau(1,1,0{:}1,1,1)=12\cdot\tau(r_{n-1}).$$

Therefore, we have $\tau(r_n)\geqq 3\cdot\tau(r_{n-1})$. Because $\tau(r_1)=3$, we have $\tau(r_n)\geqq 3^n$. On the other hand, by the distributive law, it is easy to see that $\tau(r_n)\leqq 3^n$. Hence the theorem. (Q.E.D)

4. Experimental Results

As for the simplification of AND-EXOR expressions, various methods have been developed for each class of the expressions. They can be divided into two classes: One uses spectral techniques, the other one iterative improvement techniques. Spectral methods normally work on complete truth tables, requiring 2^n entries. An exception is [BES 91] which processes cubes directly to simplify ESOPs.

Minimization algorithms for FPRME, PSDRME, KRO, and PSDKRO are known. For example, to minimize PSDKRO, the space and computation time of $O(3^n)$ and $O(n\cdot 3^n)$, respectively, are sufficient[DAV76, LUI90, MUK90]. It is not difficult to obtain the minimum of 14-variable functions using a workstation. On the other hand, the iterative improvement method reduces the number of products by modifying the set of product terms in the expressions. The necessary memory size is proportional to $n\cdot t$, where t is the number of products and n is the number of the variables. The computation time is proportional to t^2 or t^3. By this method, we can simplify ESOPs, the most general AND-EXOR expressions, but it takes much computation time and cannot guarantee the minimality of the solutions [BRA90, EVE67, FLE87, PER89, HEL88, ROB82, SAS90a, SAS90b]. For each classes of AND-EXOR expressions, we developed an optimization program, and minimized various functions.

4.1 Arithmetic functions

Table 4.1 compares the numbers of products and literals of SOPs and ESOPs for various arithmetic functions. The columns headed with "1-bit" denote the ESOPs with two-valued input variables, and the columns headed with "2-bit" denote the ESOPs with four-valued inputs. They correspond to PLAs with 1-bit decoders and PLAs with two-bit decoders, respectively [SAS 81, SAS84, SAS90b, PER89]. In these examples, ESOPs require fewer products than SOPs. In this experiment SOPs were minimized by QM[SAS84], and ESOPs were simplified by a non-deterministic algorithm [BRA91].

Table 4.2 compares the number of products for various AND-EXOR expressions. |f| denotes the number of products in the truth tables. The number of products tends to decrease in the following order: |f|, PPRME, FPRME, SOP, KRO, PSDRME, PSDKRO, and ESOP.

4.2 Randomly generated functions (n=4 to 14)

Table 4.3 shows the number of products for randomly generated functions. For each n, a pseudo-random function of n variables with 2^{n-1} minterms were generated, and minimized. In this case, the numbers of products for |f| and PPRME are comparable, but as for FPRME, KRO, PSDRME, PSDKRO, SOP, and ESOP, the numbers of products decrease in this order. In this experiment SOPs were simplified by MINI2 [SAS84], and ESOPs were simplified by EXMIN [SAS90b].

4.3 4-variable functions

There are 65536 functions of 4 variables. These functions can be classified into 402 equivalence classes under NP equivalence relation [HAR65, MUR79]. Table 4.4 shows the distribution of the number of products for the functions. This result was obtained by minimizing each of 402 representative functions, and then weighting by the number of the different functions in each equivalence class. From this table, we can obtain the average number of the products. In the table, t denotes the number of products and av denotes the average number of the products. The average numbers of the products decreases in the following order: FPRME, KRO, PSDRME, SOP, PSDKRO, and ESOP. In this experiment we optimized SOPs by QM, and ESOPs by an exhaustive method [KOD89].

4.4 5-variable functions

There are 2^{32} different 5-variable functions. These functions can be classified into 6936 equivalence classes under LP equivalence relation [SAS91, KOD91]. Table 4.5 shows the distribution of the number of products for the functions. This result was obtained by optimizing each of 6936 representative functions, and then multiplying the number of the functions in each class. From this table, we can calculate the average numbers of the products for KRO, PSDRME, PSDKRO, and ESOP. The average numbers of the products decrease in this order. In this experiment the ESOPs were optimized by a special minimization algorithm [KOD90].

4.5 6-variable functions

It has been verified that arbitrary 6-variable function can be realized by ESOPs with at most 16 products [KOD91]. So, we have the following result.

<u>Theorem 4.1</u> : Let $\Phi(n)$ and $\Psi(n)$ be the sufficient number of products to realize an arbitrary n-variable function by an SOP and an ESOP, respectively. Then $\Phi(n)=2^{n-1}$ and $\Psi(n)=2^{n-2}$ for $n \geqq 6$.

Table 4.1. Number of products and literals to represent arithmetic functions.

Data	# of products				# of literals			
	SOP		ESOP		SOP		ESOP	
Name	1bit	2bit	1bit	2bit	1bit	2bit	1bit	2bit
ADR4	75	17	31	11	423	139	168	99
LOG8	123	98	96	94	1019	1162	785	1090
MLP4	121	85	61	52	889	910	441	467
NRM4	120	70	73	56	887	799	602	618
RDM8	76	52	31	26	406	431	181	208
ROT8	57	38	35	28	389	414	280	353
SQR8	180	147	114	112	1398	1675	809	1181
WGT8	255	54	54	25	2078	530	356	207

Table 4.2. Number of products to realize arithmetic functions.

Data Name	|f|	PP RME	FP RME	PSD RME	KRO	PSD KRO	ESOP	SOP
ADR4	255	34	34	34	34	34	31	75
LOG8	255	253	193	163	171	128	96	123
MLP4	225	97	97	90	97	81	61	121
NRM4	255	216	185	150	157	105	71	120
RDM8	255	56	56	46	56	41	31	76
ROT8	255	225	118	81	83	44	35	57
SQR8	255	168	168	164	168	146	112	180
WGT8	255	107	107	107	107	107	54	255
SYM9	420	210	173	127	173	90	52	84

Table 4.3. Number of products to realize randomly generated functions.

n	|f|	PP RME	FP RME	KRO	PSD RME	PSD KRO	ESOP	SOP
4	8	6	5	4	4	4	3	4
5	16	16	10	8	7	6	5	6
6	32	36	17	17	13	12	10	13
7	64	64	54	48	30	26	19	24
8	128	122	101	100	56	50	39	46
9	256	236	226	212	112	99	69	86
10	512	528	459	439	235	206	142	167
11	1024	1021	956	925	458	391	276	331
12	2048	1996	1925	1899	909	775	572	611
13	4096	4136	3923	3865	1813	1563	1097	1157
14	8192	8210	7924	7826	3617	3107	2190	2234

Table 4.4. Number of 4-variable functions requiring t products.

t	FPRME	KRO	PSDRME	PSDKRO	ESOP	SOP
0	1	1	1	1	1	1
1	81	81	81	81	81	81
2	836	2268	1764	2268	2268	1804
3	3496	8424	11864	18248	21744	13472
4	8878	15174	27934	33910	37530	28904
5	17884	19260	19628	9708	3888	17032
6	20152	19440	3880	1296	24	3704
7	11600	864	360	0		512
8	2336	0	24	24		26
9	240	0				
10	32	24				
av	5.50	4.73	4.20	3.84	3.66	4.13

Table 4.5 Number of the 5-variable functions requiring t products.

t	KRO	PSDRME	PSDKRO	ESOP
0	1	1	1	1
1	243	243	243	243
2	24948	1620	24948	24947
3	354780	345060	1346220	1351836
4	2508570	3333906	16417026	39365190
5	12029418	28341090	170332794	545193342
6	55321704	222639840	828743400	2398267764
7	187202664	1237084812	2280973932	1299295404
8	418029660	1489676400	883268712	11460744
9	804890520	879161364	104197428	7824
10	1006381476	345677544	9049320	
11	1053603288	79186896	587088	
12	544903200	7718328	26136	
13	195821712	1632960	0	
14	13630680	155520	0	
15	256608	11664	0	
16	0	48	48	
17	7776			
21	48			
av	10.05	8.01	6.97	6.17

5. Conclusion

In this paper, we presented 7 classes of AND-EXOR expressions: PPRME, FPRME, PSDRME, GRME, KRO, PSDKRO, and ESOP. In particular, PSDRME is a new class defined in this paper. Also we developed optimization programs for each class and optimized various functions. ESOPs require the least number of products but are difficult to minimize. PSDKROs require the least but one number of products. The space and time complexity for the optimization of PSDKRO are both $O(3^n)$. Also we showed that the ESOP for the function $x_1y_1 \vee x_2y_2 \vee \cdots \vee x_ny_n$ requires 2^n-1 products.

Acknowledgements

This work was supported in part by a Grant in Aid for Scientific Research of the Ministry of Education, Science and and Culture of Japan. The author thanks Mr.N.Koda and Dr. D. Brand for their programs. Prof.P.Besslich carefully read this paper. Profs. M.Davio, M.R. Mukerjee and J. Muzio sent me useful papers. The discussion with Prof.M.Perkowski was quite useful for the improvement of the paper. Prof.Machida gave me a chance to write this paper. Prof.S.Iwata's comment [IWA 91] was the key to the proof of the Theorems 3.3.

References

[BRA 91] D.Brand and T.Sasao, "On the minimization of AND-EXOR expressions",International Workshop on Logic Synthesis, Research Triangle Park, NC, May 1991.

[BES 83] Ph. W. Besslich,"Efficient computer method for ExOR logic design", IEE Proc., vol.130, Part E, pp.203-206, 1983.

[BES 85] Ph. W. Besslich,"Spectral processing of switching functions using signal-flow transformations", in M.Karpovsky (Ed.), Spectral Techniques and Fault Detection, Orlando, FL:Academic Press, 1985, pp.91-141.

[BES 91] Ph. W. Besslich and M.W.Riege,"An efficient program for logic synthesis of mod-2 sums expressions"EuroASIC, Pris/France, May 1991, Proc. IEEE Compu.Soc. Press.

[BIO 73] G.Bioul, M.Davio and J.P.Deschamps, "Minimization of ring-sum expansions of Boolean functions", Philips Res.Rpts., vol.28, pp. 17-36, 1973.

[COH 62] M.Cohn, "Inconsistent canonical forms of switching functions", IRE Trans. on Elec. Computers, Vol.EC-11, pp.284-285, April 1962.

[DAV 78] M.Davio, J-P Deschamps, and A. Thayse, "Discrete and switching functions", McGraw-Hill International, 1978.

[DAM 90] C.Damm, "Lower bounds on the size of parity normal forms computing clique functions",Fachbereich Informatik, Humboldt-Universitat Berlin, 1990.

[EVE 67] S.Even, I.Kohavi and A.Paz,"On minimal modulo-2 sum of products for switching functions", IEEE Trans. on Electron. Computers, Vol.EC-16, pp.671-674, Oct. 1967.

[FLE 87] H.Fleisher, M.Tavel and J.Yeager,"A computer algorithm for minimizing Reed-Muller canonical forms", IEEE Trans. on Computers Vol. C-36, No.2 Feb. 1987.

[GRE 76] D.H.Green and I.S.Taylor:"Multiple-valued switching circuit design by means of generalized Reed-Muller expansions",Digital Processes, vol.2, pp.63-81, 1976.

[IWA 91] S.Iwata, private communication, June 13,1991.

[HAR 65] M.A.Harrison, Introduction to Switching and Automata Theory, McGraw-Hill, 1965.

[HAS 86] J.T.Hastad,"Computational limitations for small-depth circuits" The MIT Press, 1987.

[HEL 88] 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.

[HON 74] 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.

[KAR 72] R.M.Karp, "Reducibility among combinatorial problems", in Complexity of Computer Computations (R.E.Miller and J.W.Thatcher eds.) Plenum Press, New York, 1978, pp.85-104.

[KOD 89] N.Koda and T.Sasao,"Four-variable AND-EXOR minimum expressions and their properties"(in Japanese) IEICE Technical Report, FTS89-25, Oct. 24, 1989.

[KOD 90] N.Koda and T.Sasao,"On the minimization of 5-variable AND-EXOR expressions",(in Japanese), National Convention of IEICE Japan, SA-3-3, Oct. 1990.

[KOD 91] N.Koda and T.Sasao, "On the number of product terms of AND-EXOR minimum expressions",(in Japanese) IEICE Technical Report, FTS91-22, July, 1991.

[LUI 90] P.K.Lui and J.Muzio,"Boolean Matrix transforms for the parity spectrum and the minimization of modulo-2 canonical expansions", Department of Computer Science, University of Victoria, DCS-135-IR, July 1990.

[MUK 70] A. Mukhopadhyay and G.Schmitz,"Minimization of Exclusive OR and logical Equivalence of switching circuits", IEEE Trans. Comput., C-19, pp.132-140, 1970.

[MUK 90] M.R.Mukerjee, "Minimization of ring-sum expansion of mixed polarity", AMSE Symp. on Modeling and Simulation, Greensboro, Oct.1990.

[MUL 54] D.E.Muller,"Application of Boolean algebra to switching circuit design and to error detection", IRE Trans. Electron. Comput., EC-3,pp. 6-12, 1954.

[MUR 79] S.Muroga, Logic Design and Switching Theory, John Wiley & Sons, 1979.

[PAP 79] G.Papakonstantinou,"Minimization of modulo-2 sum of products", IEEE Trans. Comput., C-28, pp.163-167, 1979.

[PER 89] M.Perkowski, M.Helliwell, and P.Wu,"Minimization of multiple-valued input multi-output mixed-radix exclusive sum of products for incompletely specified Boolean functions", ISMVL-89, pp.256-263, May 1989.

[PER 90] M.Perkowski and M.Chrzanowska-Jeske", An exact algorithm to

minimize mixed-radix exclusive sums of products for incompletely specified Boolean functions", Proc.ISCAS, pp.1652-1655, June 1990.
[RAZ 86] A.Razborov,"Lower bounds on the size of bounded depth circuits over the basis {$\wedge, \oplus$}", Preprint Steklov Inst. for Math.,Moscow 1986 (see also Mat. Zam 41 (1987), 598-607)(in Russian).
[RED 72] S.M.Reddy, "Easily testable realization for logic functions", IEEE Trans. Comput., C-21, pp.1083-1088, 1972.
[REE 54] I.S.Reed,"A class of multiple-error-correcting codes and the decoding scheme", IRE Trans.Information Theory PGIT-4. pp.38-49, 1954.
[ROB 82] J.P.Robinson and Chia-Lung Yeh, "A method for modulo-2 minimization", IEEE Trans. Comput., C-31, pp.800-801, 1982.
[SAL 79] K.K.Saluja and E.H.Ong, "Minimization of Reed-Muller canonic expansion", IEEE Trans. Comput., C-28, pp.535-537, 1979.
[SAS 81] T.Sasao, "Multiple-valued decomposition of generalized Boolean functions and the complexity of programmable logic arrays", IEEE Trans. on Comput. vol.C-30, No.9, pp.635-643, Sept.1981.
[SAS 84] T.Sasao,"Input variable assignment and output phase optimization of PLA's", IEEE Trans. on Comput. vol C-33, No.10, pp.879-894, Oct. 1984.
[SAS 90a] T.Sasao and P.Besslich,"On the complexity of MOD-2 Sum PLA's", IEEE Trans. on Comput. Vol.39. No.2, pp.262-266, Feb.1990.
[SAS 90b]T.Sasao, "EXMIN: A simplification algorithm for Exclusive-OR-Sum-of-Products Expressions for multiple-Valued input two-valued output functions", ISMVL-90, May 1990, pp.128-135.
[SAS 90d] T.Sasao,"Exclusive-or sum-of-products expressions:their properties and a minimization algorithm", IEICE Technical Report,VLD-90-89,Dec.1990.
[SAS 91] T.Sasao,"A transformation of multiple-valued input two-valued output functions and its application to simplification of exclusive-or sum-of-products expressions", ISMVL-91, May 1991, pp.270-279.
[SAV 76] J.E.Savage, "The complexity of Computing", John Wiley, New York, 1976.
[WEG 87] I.Wegener, "The complexity of Boolean functions", John Wiley & Sons, New York, 1987.

APPENDIX

(Proof for Lemma 3.5) Let F be an MESOP for f and be represented as follows:

$F(0)x^0 \oplus F(1)x^1 \oplus F(2)x^2$, ----------(A1)

where $F(a)$ ($a \in \{0,1,2\}$) are ESOPs which do not contain variable x, $x^0 = \bar{x}$, $x^1 = x$, and $x^2=1$.

By setting $x=0$ in (A1), $F(0) \oplus F(2) = f(0)$. ------------(A2)

By setting $x=1$ in (A1), $F(1) \oplus F(2) = f(1)$. ------------(A3)

By (A2)$\oplus$(A3), we have $F(0) \oplus F(1) = f(0) \oplus f(1)$. ------------(A4)

Let $\tau(a)=\tau(F(a))$. From (A4), we have $\tau(0)+\tau(1) \geqq \tau(f(0) \oplus f(1))$. Note that $\tau(f)=\tau(0)+\tau(1)+\tau(2)$. Because $\tau(2) \geqq 0$, we have $\tau(f) \geqq \tau(f(0) \oplus f(1))$. (Q.E.D.)

(Proof for Lemma 3.6) Let F be an MESOP for f and be represented as follows:

$$F(0,0)x^0y^0 \oplus F(0,1)x^0y^1 \oplus F(0,2)x^0y^2 \oplus F(1,0)x^1y^0 \oplus F(1,1)x^1y^1$$
$$\oplus F(1,2)x^1y^2 \oplus F(2,0)x^2y^0 \oplus F(2,1)x^2y^1 \oplus F(2,2)x^2y^2, \quad \text{-------}(B1)$$

where $F(a,b)$ $(a,b \in \{0,1,2\})$ are ESOPs which do not contain variable x nor y, $x^0 = \bar{x}$, $x^1 = x$, and $x^2=1$, $y^0 = \bar{y}$, $y^1 = y$, and $y^2=1$.

By setting $(x,y) = (0,0)$ in (B1),

$$F(0,0) \oplus F(0,2) \oplus F(2,0) \oplus F(2,2) = f(0,0). \quad \text{------}(B2)$$

By setting $(x,y) = (1,1)$ in (B1),

$$F(1,1) \oplus F(1,2) \oplus F(2,1) \oplus F(2,2) = f(1,1). \quad \text{------}(B3)$$

By setting $(x,y) = (0,1)$ in (B1),

$$F(0,1) \oplus F(0,2) \oplus F(2,1) \oplus F(2,2) = f(0,1). \quad \text{------}(B4)$$

By setting $(x,y) = (1,0)$ in (B1),

$$F(1,0) \oplus F(1,2) \oplus F(2,0) \oplus F(2,2) = f(1,0). \quad \text{------}(B5)$$

By (B2) and (B4), $F(0,0) \oplus F(0,1) \oplus F(2,0) \oplus F(2,1) = f(0,0) \oplus f(0,1)$ --(B6)

By (B2) and (B5), $F(0,0) \oplus F(0,2) \oplus F(1,0) \oplus F(1,2) = f(0,0) \oplus f(1,0)$ --(B7)

By (B3) and (B4), $F(0,1) \oplus F(0,2) \oplus F(1,1) \oplus F(1,2) = f(1,1) \oplus f(0,1)$ --(B8)

By (B3) and (B5), $F(1,0) \oplus F(1,1) \oplus F(2,0) \oplus F(2,1) = f(1,1) \oplus f(1,0)$ --(B9)

Let $\tau(a,b) = \tau(F(a,b))$. From (B6) to (B9), we have

$\tau(0,0) + \tau(0,1) + \tau(2,0) + \tau(2,1) \geqq \tau(0,0:0,1)$,
$\tau(0,0) + \tau(0,2) + \tau(1,0) + \tau(1,2) \geqq \tau(0,0:1,0)$,
$\tau(0,1) + \tau(0,2) + \tau(1,1) + \tau(1,2) \geqq \tau(1,1:0,1)$, and
$\tau(1,0) + \tau(1,1) + \tau(2,0) + \tau(2,1) \geqq \tau(1,1:1,0)$.

By adding thefour inequations above, we have

$$2\{\tau(0,0) + \tau(0,1) + \tau(0,2) + \tau(1,0) + \tau(1,1) + \tau(1,2) + \tau(2,0) + \tau(2,1)\} \geqq$$
$$\tau(0,0:0,1) + \tau(0,0:1,0) + \tau(1,1:0,1) + \tau(1,1:1,0).$$

Note that $\tau(f) = \tau(0,0) + \tau(0,1) + \tau(0,2) + \tau(1,0) + \tau(1,1) + \tau(1,2) + \tau(2,0) + \tau(2,1) + \tau(2,2)$. Because $\tau(2,2) \geqq 0$, we have $2 \cdot \tau(f) \geqq \tau(0,0:0,1) + \tau(0,0:1,0) + \tau(1,1:0,1) + \tau(1,1:1,0)$. Hence the lemma.

(Q.E.D.)

(Proof for Lemma 3.7) Let F be an MESOP for f and be represented as follows:

$$F(0,0,0)x^0y^0z^0 \oplus F(0,0,1)x^0y^0z^1 \oplus F(0,0,2)x^0y^0z^2 \oplus F(0,1,0)x^0y^1z^0$$
$$\oplus F(0,1,1)x^0y^1z^1 \oplus F(0,1,2)x^0y^1z^2 \oplus F(0,2,0)x^0y^2z^0 \oplus F(0,2,1)x^0y^2z^1$$
$$\oplus F(0,2,2)x^0y^2z^2 \oplus F(1,0,0)x^1y^0z^0 \oplus F(1,0,1)x^1y^0z^1 \oplus F(1,0,2)x^1y^0z^2$$
$$\oplus F(1,1,0)x^1y^1z^0 \oplus F(1,1,1)x^1y^1z^1 \oplus F(1,1,2)x^1y^1z^2 \oplus F(1,2,0)x^1y^2z^0$$
$$\oplus F(1,2,1)x^1y^2z^1 \oplus F(1,2,2)x^1y^2z^2 \oplus F(2,0,0)x^2y^0z^0 \oplus F(2,0,1)x^2y^0z^1$$
$$\oplus F(2,0,2)x^2y^0z^2 \oplus F(2,1,0)x^2y^1z^0 \oplus F(2,1,1)x^2y^1z^1 \oplus F(2,1,2)x^2y^1z^2$$
$$\oplus F(2,2,0)x^2y^2z^0 \oplus F(2,2,1)x^2y^2z^1 \oplus F(2,2,2)x^2y^2z^2, \quad \text{---}(C1)$$

where $F(a,b,c)$ $(a,b,c \in \{0,1,2\})$ are ESOPs which do not contain variable x, y nor z,

$x^0 = \bar{x}$, $x^1 = x$, $x^2=1$, $y^0 = \bar{y}$, $y^1 = y$, $y^2=1$, $z^0 = \bar{z}$, $z^1 = z$, and $z^2=1$.

By setting $(x,y,z) = (0,0,0)$ in (C1),

$$F(0,0,0) \oplus F(0,0,2) \oplus F(0,2,0) \oplus F(0,2,2) \oplus F(2,0,0) \oplus F(2,0,2) \oplus F(2,2,0)$$
$$\oplus F(2,2,2) = f(0,0,0). \quad \text{-------}(C2)$$

By setting $(x,y,z) = (0,1,1)$ in (C1),

$$F(0,1,1) \oplus F(0,1,2) \oplus F(0,2,1) \oplus F(0,2,2) \oplus F(2,1,1) \oplus F(2,1,2) \oplus F(2,2,1)$$

$\oplus F(2,2,2) = f(0,1,1)$. ----------(C3)

By setting $(x,y,z)=(1,0,1)$ in (C1),

$F(1,0,1)\oplus F(1,0,2)\oplus F(1,2,1)\oplus F(1,2,2)\oplus F(2,0,1)\oplus F(2,0,2)\oplus F(2,2,1)$
$\oplus F(2,2,2) = f(1,0,1)$. ------------(C4)

By setting $(x,y,z)=(1,1,0)$ in (C1),

$F(1,1,0)\oplus F(1,1,2)\oplus F(1,2,0)\oplus F(1,2,2)\oplus F(2,1,0)\oplus F(2,1,2)\oplus F(2,2,0)$
$\oplus F(2,2,2) = f(1,1,0)$. ------------(C5)

By setting $(x,y,z)=(0,0,1)$ in (C1),

$F(0,0,1)\oplus F(0,0,2)\oplus F(0,2,1)\oplus F(0,2,2)\oplus F(2,0,1)\oplus F(2,0,2)\oplus F(2,2,1)$
$\oplus F(2,2,2) = f(0,0,1)$. ------------(C6)

By setting $(x,y,z)=(0,1,0)$ in (C1),

$F(0,1,0)\oplus F(0,1,2)\oplus F(0,2,0)\oplus F(0,2,2)\oplus F(2,1,0)\oplus F(2,1,2)\oplus F(2,2,0)$
$\oplus F(2,2,2) = f(0,1,0)$. ------------(C7)

By setting $(x,y,z)=(1,0,0)$ in (C1),

$F(1,0,0)\oplus F(1,0,2)\oplus F(1,2,0)\oplus F(1,2,2)\oplus F(2,0,0)\oplus F(2,0,2)\oplus F(2,2,0)$
$\oplus F(2,2,2) = f(1,0,0)$. ------------(C8)

By setting $(x,y,z)=(1,1,1)$ in (C1),

$F(1,1,1)\oplus F(1,1,2)\oplus F(1,2,1)\oplus F(1,2,2)\oplus F(2,1,1)\oplus F(2,1,2)\oplus F(2,2,1)$
$\oplus F(2,2,2) = f(1,1,1)$. ------------(C9)

By (C2)$\oplus$(C6),

$F(0,0,0)\oplus F(0,0,1)\oplus F(0,2,0)\oplus F(0,2,1)\oplus F(2,0,0)\oplus F(2,0,1)\oplus F(2,2,0)$
$\oplus F(2,2,1)=f(0,0,0)\oplus f(0,0,1)$. ----(C10)

By (C2)$\oplus$(C7),

$F(0,0,0)\oplus F(0,0,2)\oplus F(0,1,0)\oplus F(0,1,2)\oplus F(2,0,0)\oplus F(2,0,2)\oplus F(2,1,0)$
$\oplus F(2,1,2)=f(0,0,0)\oplus f(0,1,0)$. -----(C11)

By (C2)$\oplus$(C8),

$F(0,0,0)\oplus F(0,0,2)\oplus F(0,2,0)\oplus F(0,2,2)\oplus F(1,0,0)\oplus F(1,0,2)\oplus F(1,2,0)$
$\oplus F(1,2,2)=f(0,0,0)\oplus f(1,0,0)$. -----(C12)

By (C3)$\oplus$(C6),

$F(0,0,1)\oplus F(0,0,2)\oplus F(0,1,1)\oplus F(0,1,2)\oplus F(2,0,1)\oplus F(2,0,2)\oplus F(2,1,1)$
$\oplus F(2,1,2)=f(0,1,1)\oplus f(0,0,1)$. -----(C13)

By (C3)$\oplus$(C7),

$F(0,1,0)\oplus F(0,1,1)\oplus F(0,2,0)\oplus F(0,2,1)\oplus F(2,1,0)\oplus F(2,1,1)\oplus F(2,2,0)$
$\oplus F(2,2,1)=f(0,1,1)\oplus f(0,1,0)$. -----(C14)

By (C3)$\oplus$(C9),

$F(0,1,1)\oplus F(0,1,2)\oplus F(0,2,1)\oplus F(0,2,2)\oplus F(1,1,1)\oplus F(1,1,2)\oplus F(1,2,1)$
$\oplus F(1,2,2)=f(0,1,1)\oplus f(1,1,1)$. -----(C15)

By (C4)$\oplus$(C6),

$F(0,0,1)\oplus F(0,0,2)\oplus F(0,2,1)\oplus F(0,2,2)\oplus F(1,0,1)\oplus F(1,0,2)\oplus F(1,2,1)$
$\oplus F(1,2,2)=f(1,0,1)\oplus f(0,0,1)$. -----(C16)

By (C4)$\oplus$(C8),

$F(1,0,0)\oplus F(1,0,1)\oplus F(1,2,0)\oplus F(1,2,1)\oplus F(2,0,0)\oplus F(2,0,1)\oplus F(2,2,0)$
$\oplus F(2,2,1)=f(1,0,1)\oplus f(1,0,0)$. -----(C17)

By (C4)$\oplus$(C9),

$F(1,0,1)\oplus F(1,0,2)\oplus F(1,1,1)\oplus F(1,1,2)\oplus F(2,0,1)\oplus F(2,0,2)\oplus F(2,1,1)$
$\oplus F(2,1,2)=f(1,0,1)\oplus f(1,1,1)$. -----(C18)

By (C5)$\oplus$(C7),

$F(0,1,0)\oplus F(0,1,2)\oplus F(0,2,0)\oplus F(0,2,2)\oplus F(1,1,0)\oplus F(1,1,2)\oplus F(1,2,0)$
$\oplus F(1,2,2)=f(1,1,0)\oplus f(0,1,0)$. -----(C19)

By (C5)$\oplus$(C8),

$F(1,0,0)\oplus F(1,0,2)\oplus F(1,1,0)\oplus F(1,1,2)\oplus F(2,0,0)\oplus F(2,0,2)\oplus F(2,1,0)$
$\oplus F(2,1,2)=f(1,1,0)\oplus f(1,0,0)$. -----(C20)

By (C5)$\oplus$(C9),

$F(1,1,0)\oplus F(1,1,1)\oplus F(1,2,0)\oplus F(1,2,1)\oplus F(2,1,0)\oplus F(2,1,1)\oplus F(2,2,0)$
$\oplus F(2,2,1)=f(1,1,0)\oplus f(1,1,1)$. ------(C21)

Let $\tau(a,b,c)=\tau(F(a,b,c))$, and $\tau(a,b,c:d,e,h)=\tau(f(a,b,c)\oplus f(d,e,h))$.
From (C10) to (C21), we have $3\tau(0,0,0)+3\tau(0,0,1)+4\tau(0,0,2)+3\tau(0,1,0)$
$+3\tau(0,1,1)+4\tau(0,1,2)+4\tau(0,2,0)+4\tau(0,2,1)+4\tau(0,2,2)+3\tau(1,0,0)$
$+3\tau(1,0,1)+4\tau(1,0,2)+3\tau(1,1,0)+3\tau(1,1,1)+4\tau(1,1,2)+4\tau(1,2,0)$
$+4\tau(1,2,1)+4\tau(1,2,2)+4\tau(2,0,0)+4\tau(2,0,1)+4\tau(2,0,2)+4\tau(2,1,0)$
$+4\tau(2,1,1)+4\tau(2,1,2)+4\tau(2,2,0)+4\tau(2,2,1)\geqq$
$\{\tau(0,0,0:0,0,1)+\tau(0,0,0:0,1,0)+\tau(0,0,0:1,0,0)+\tau(0,1,1:0,0,1)$
$+\tau(0,1,1:0,1,0)+\tau(0,1,1:1,1,1)+\tau(1,0,1:0,0,1)+\tau(1,0,1:1,0,0)$
$+\tau(1,0,1:1,0,0)+\tau(1,1,0:0,1,0)+\tau(1,1,0:1,0,0)+\tau(1,1,0:1,1,1)\}$.
Note that $\tau(f)=\tau(0,0,0)+\tau(0,0,1)+\tau(0,0,2)+\tau(0,1,0)+\tau(0,1,1)$
$+\tau(0,1,2)+\tau(0,2,0)+\tau(0,2,1)+\tau(0,2,2)+\tau(1,0,0)+\tau(1,0,1)+\tau(1,0,2)$
$+\tau(1,1,0)+\tau(1,1,1)+\tau(1,1,2)+\tau(1,2,0)+\tau(1,2,1)+\tau(1,2,2)+\tau(2,0,0)$
$+\tau(2,0,1)+\tau(2,0,2)+\tau(2,1,0)+\tau(2,1,1)+\tau(2,1,2)+\tau(2,2,0)+\tau(2,2,1)$
$+\tau(2,2,2)$.
Because $\tau(a,b,c)\geqq 0$, we have $4\cdot\tau(f)\geqq\{\tau(0,0,0:0,0,1)+\tau(0,0,0:0,1,0)$
$+\tau(0,0,0:1,0,0)+\tau(0,1,1:0,0,1)+\tau(0,1,1:0,1,0)+\tau(0,1,1:1,1,1)$
$+\tau(1,0,1:0,0,1)+\tau(1,0,1:1,0,0)+\tau(1,0,1:1,1,1)+\tau(1,1,0:0,1,0)$
$+\tau(1,1,0:1,0,0)+\tau(1,1,0:1,1,1)\}$. Hence the lemma. (Q.E.D)