設 計 自 動 化 79－3
（1996. 2. 9）

符号化したMRCFによるAND-EXOR論理式の厳密最小化

越智 裕之

広島市立大学 情報科学部 情報工学科

〒731-31 広島市安佐南区沼田町大塚 151-5

ochi@ce.hiroshima-cu.ac.jp

本稿ではAND-EXOR論理式(ESOP)の厳密最小化を二分決定グラフ上で効率よく行なうための手法として、Encoded MRCF (EMRCF)を用いた手法を提案する。これは、笹尾がHelliwell関数を一般化、改良して提案したMRCFに基づく手法で構成される二分決定グラフのサイズが大きくならないよう、MRCFに適切な符号化を施したものである。

Sun SPARC station 10 (主記憶192MB)上で行なった実験では、実験した19個の5～9変数論理関数全てについて、従来のMRCFに基づく方法に比べて必要なメモリ領域が削減され、最も良い例では1/3に削減された。

An Exact Minimization of AND-EXOR Expressions Using Encoded MRCF

Hiroyuki OCHI

Department of Computer Engineering
Faculty of Information Sciences
Hiroshima City University

Asaminami-Ku, Hiroshima, 731-31, JAPAN

In this paper, an exact-minimization method for an AND-EXOR expressions (ESOP) based on an EMRCF is proposed. Sasao proposed an MRCF-based method for ESOP minimization as an improvement of the method based on a Helliwell function, and EMRCF is an MRCF applying a novel encoding so that size of BDDs for MRCF is reduced.

The proposed algorithm is implemented and evaluated on Sun SPARC station 10 with 192MB main memory. From experimental results, it is shown that required space is reduced to 1/3 in the best case compared with the conventional method based on MRCF.

1 Introduction

Utilizing *exclusive-OR (EXOR)* gates, logic networks can be implemented with fewer number of gates and interconnections compared with AND-OR networks. For example, arithmetic, telecommunication, and error correcting circuits are realized efficiently by utilizing EXOR gates. To enable efficient EXOR-based design, development of CAD tools that utilize EXOR gates is indispensable.

Arbitrary product terms combined by EXORs is called an *exclusive-or sum-of-products expression (ESOP)*. Various studies have been made for exact- and quasi-minimization of number of product terms of an ESOP for a given Boolean function, including [1]. Recently, Perkowski et al. [2] formulated the problem by using a *Helliwell equation*. Let n be the number of input variables of a given Boolean function, a Helliwell equation is an equation of 3^n Boolean variables, and a minimum ESOP for the given Boolean function corresponds to a solutions of the Helliwell equation with minimum number of 1's. Sasao [3] proposed a method based on a *reduced covering function (RCF)*, a generalization of the Helliwell function, to solve greater problems by assuming that exact minimum ESOPs for r-variable Boolean functions are known for a small constant r, e. g., 1, 2, 3, and 4. To implement the algorithm efficiently using binary decision diagrams (BDDs) [4], a *modified reduced covering function (MRCF)* was also developed.

In this paper an *encoded modified reduced covering function (EMRCF)* is proposed to solve greater problems by reducing the required space to generate a BDD for MRCF. From experimental results, 6-, 7-, 8-, and 9-variable Boolean functions of up to respectively 10, 9, 8, and 8 product terms in ESOP are minimized within 192 MB main memory, and shown that required space is reduced compared with the conventional method based on MRCF.

2 Preliminary

2.1 Reduced Covering Function (RCF)

Def. 1 Let r be a non-negative integer constant, $x = \langle x_n, \ldots, x_1 \rangle$ be Boolean variables, and $f(x)$ be an n-variable Boolean function ($n \geq r$).

Def. 2 Let $a = \langle a_n, \ldots, a_{r+1} \rangle$. Let $x_i^{a_i}$ ($r+1 \leq i \leq n$) be literals $\bar{x}_i$, x_i, and 1, if a_i are 0, 1, and 2, respectively.

Let us consider the following ternary expansion of f :

$$f(x) = \bigoplus_{a \in \{0,1,2\}^{n-r}} (x_n^{a_n} \wedge \cdots \wedge x_{r+1}^{a_{r+1}} \wedge g_a(x_r, \ldots, x_1)), \qquad (1)$$

where g_a ($a \in \{0,1,2\}^{n-r}$) is an r-variable Boolean function.

Def. 3 Let $p = \langle p_n, \ldots, p_{r+1} \rangle$, and $b = \langle b_r, \ldots, b_1 \rangle$. Let $g = \langle g_a(x_r, \ldots, x_1) \mid a \in \{0,1,2\}^{n-r} \rangle$ be a tuple of 3^{n-r} r-variable Boolean functions. A *Reduced Covering Function (RCF)* [3] $R(g)$ is a Boolean function of g defined as follows :

$$R(g) = \bigwedge_{p \in \{0,1\}^{n-r}} \bigwedge_{b \in \{0,1\}^r} \left(\overline{f(p,b) \oplus \bigoplus_{a \in P_p} g_a(b)} \right), \qquad (2)$$

where $P_p = \{a \in \{0,1,2\}^{n-r} \mid r+1 \leq \forall i \leq n.(a_i = 2 \vee a_i = p_i)\}$.

Intuitively, P_p represents a set of cubes that cover a cube represented by p.

Proposition 1 Equation (1) holds for all possible combinations of values for $x \in \{0,1\}^n$, iff a tuple g of r-variable Boolean functions satisfy an equation $R(g) = 1$.

Example 1 When $r = 0$, $R(g)$ is equivalent to the Helliwell function [2].

Def. 4 Let $\phi(g_a)$ be a minimum ESOP for an r-variable Boolean function g_a, and $\tau(g_a)$ be the number of product terms of $\phi(g_a)$. Let

$$\tau_\Sigma(g) = \sum_{a \in \{0,1,2\}^{n-r}} \tau(g_a).$$

It is possible to generate a complete look-up table to obtain a $\phi(g_a)$ and $\tau(g_a)$ for r up to 4. From Proposition (1), an ESOP for f is associated for every solution of the equation $R(g) = 1$ as follows :

$$\bigoplus_{a \in \{0,1,2\}^{n-r}} (x_n^{a_n} \wedge \cdots \wedge x_{r+1}^{a_{r+1}} \wedge \phi(g_a(x_r, \ldots, x_1))),$$

and the number of product terms of the above ESOP is $\tau_\Sigma(g)$.

Therefore, a minimum ESOP for f is derived by finding a solution of the equation $R(g) = 1$ that minimizes τ_Σ.

To solve the problem exhaustively using *Binary Decision Diagrams (BDDs)* [4], Sasao introduced $3^{n-r}2^r$ Boolean variables, say g_{ab} ($a \in \{0,1,2\}^{n-r}$, $b \in \{0,1\}^r$), defined as follows :

$$g_{ab} = g_a(b) \; (a \in \{0,1,2\}^{n-r},\ b \in \{0,1\}^r) \qquad (3)$$

Example 2 When $n = 3$ and $r = 1$, $R(g)$ is as follows :

$$\begin{aligned} R(g) = & \left(\overline{f(0,0,0)} \oplus g_{00,0} \oplus g_{02,0} \oplus g_{20,0} \oplus g_{22,0}\right) \\ \wedge & \left(\overline{f(0,0,1)} \oplus g_{00,1} \oplus g_{02,1} \oplus g_{20,1} \oplus g_{22,1}\right) \\ \wedge & \left(\overline{f(0,1,0)} \oplus g_{01,0} \oplus g_{02,0} \oplus g_{21,0} \oplus g_{22,0}\right) \\ \wedge & \left(\overline{f(0,1,1)} \oplus g_{01,1} \oplus g_{02,1} \oplus g_{21,1} \oplus g_{22,1}\right) \\ \wedge & \left(\overline{f(1,0,0)} \oplus g_{10,0} \oplus g_{12,0} \oplus g_{20,0} \oplus g_{22,0}\right) \\ \wedge & \left(\overline{f(1,0,1)} \oplus g_{10,1} \oplus g_{12,1} \oplus g_{20,1} \oplus g_{22,1}\right) \\ \wedge & \left(\overline{f(1,1,0)} \oplus g_{11,0} \oplus g_{12,0} \oplus g_{21,0} \oplus g_{22,0}\right) \\ \wedge & \left(\overline{f(1,1,1)} \oplus g_{11,1} \oplus g_{12,1} \oplus g_{21,1} \oplus g_{22,1}\right) \end{aligned}$$

A minimum ESOP for a given Boolean function f is derived by generating a BDD for $R(g)$ and then finding least cost path from the root node to the '1' leaf node of the BDD (the cost criteria is τ_Σ.)

BDD size for an RCF $R(g)$ become smaller if greater value for r is chosen, because the number of BDD variable $3^{n-r}2^r$ become smaller. Therefore, it is better to choose greater value for r as long as evaluation of τ_Σ is not too complex.

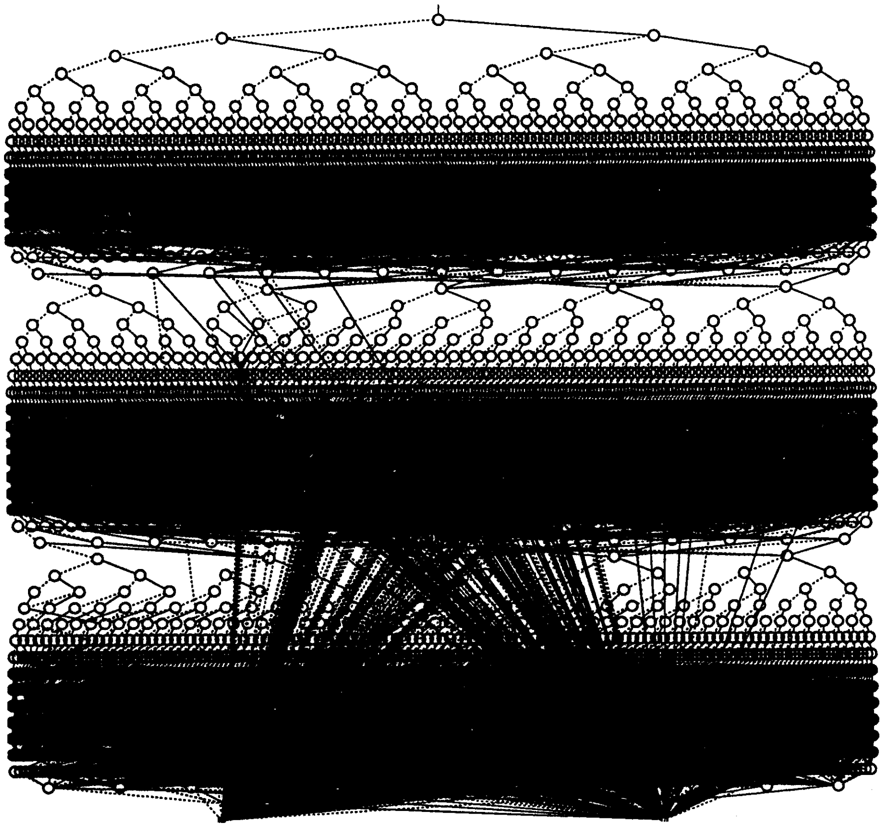

Figure 1: U_6 for $n = 5$, when $r = 4$ (25,052 nodes)

2.2 Modified Reduced Covering Function (MRCF)

If we try generating a BDD for an RCF, the size of the BDD become so large that we cannot generate it within main memory of our computer, unless n is very small. Sasao proposed to generate a BDD for a *modified reduced covering function (MRCF)* instead of an RCF.

Def. 5 A *modified reduced covering function (MRCF)* [3] is defined as $U_{t_0}(g) \wedge R(g)$, where $R(g)$ is an RCF, t_0 is the number of product terms of a near-minimum ESOP for f, and *upper bound function (UBF)* $U_{t_0}(g)$ is a Boolean function defined as follows :

$$U_{t_0}(g) = \begin{cases} 1 & \text{if } \tau_\Sigma(g) < t_0 \\ 0 & \text{otherwise} \end{cases}$$

t_0 is assumed to be derived by a "preprocessor" such as EXMIN2 [1].

Sasao also reported [3] that *0-suppressed BDDs (ZBDDs)* [5] are more efficient than conventional BDDs to solve the problem in this way.

ZBDD size for an RCF $R(g)$ become smaller if greater value is chosen for r, however, peak size of ZBDDs (maximum required space during generating a ZBDD) for an MRCF $U_{t_0}(g) \wedge R(g)$ is larger in most cases when $r = 4$ is chosen than when $r = 2$ or 3 is chosen. This seems because of the UBF $U_{t_0}(g)$ whose ZBDD become more and more complex as r and n and t_0 become greater and greater. When $r = 0$, a UBF is a symmetric threshold function and its BDD size is small. When $r = 1$, a UBF is not a threshold function, but a unate function. However, UBFs become more complex functions as r become greater. Fig. 1 shows the ZBDD for U_6 for $n = 5$ by choosing $r = 4$ (variable ordering is similar to [3].) To solve "greater" problems, it seems better to develop a new method to reduce the size of ZBDD for UBF for $r = 3$ and $r = 4$.

3 ESOP Minimization Based on EMRCF

3.1 Encoded Modified Reduced Covering Function (EMRCF)

Equation (3) provides a straight forward way to represent an r-variable Boolean function g_a by a sequence of 2^r truth values $\langle g_{ab} \mid b \in \{0,1\}^r \rangle$. In this section, an encoding method for representing g_a is proposed, which reduces drastically the size of ZBDDs for UBF when $r = 3$ and $r = 4$.

Table 1: Ordered list of all 2-variable Boolean functions and its encoding

Boolean function	Sorting keys: terms	Truth values				Encoding: ν_{G^2} in binary			
0	0	0	0	0	0	0	0	0	0
$\bar{x}_2\bar{x}_1$	1	0	0	0	1	0	0	0	1
$\bar{x}_2 x_1$	1	0	0	1	0	0	0	1	0
$\bar{x}_2$	1	0	0	1	1	0	0	1	1
$x_2\bar{x}_1$	1	0	1	0	0	0	1	0	0
$\bar{x}_1$	1	0	1	0	1	0	1	0	1
$x_2 x_1$	1	1	0	0	0	0	1	1	0
x_1	1	1	0	1	0	0	1	1	1
x_2	1	1	1	0	0	1	0	0	0
1	1	1	1	1	1	1	0	0	1
$x_2 \oplus x_1$	2	0	1	1	0	1	0	1	0
$1 \oplus x_2 x_1$	2	0	1	1	1	1	0	1	1
$x_2 \oplus \bar{x}_1$	2	1	0	0	1	1	1	0	0
$1 \oplus x_2\bar{x}_1$	2	1	0	1	1	1	1	0	1
$1 \oplus \bar{x}_2 x_1$	2	1	1	0	1	1	1	1	0
$1 \oplus \bar{x}_2\bar{x}_1$	2	1	1	1	0	1	1	1	1

Def. 6 Number $\nu_S(e)$ for an element e of an ordered set S is defined as the number of elements which precedes e in S; $\nu_S(e) = 0$, if e is the first element of S.

Def. 7 Let S_1 and S_2 be ordered sets ($S_1 \cap S_2 = \phi$.) $S = S_1 \# S_2$ is a concatenation of ordered sets S_1 and S_2; let $e_1 \in S_1$ and $e_2 \in S_2$, $\nu_S(e_1) = \nu_{S_1}(e_1)$ and $\nu_S(e_2) = \nu_{S_2}(e_2) + |S_1|$.

Def. 8 Let τ^r be the maximum number of product terms of exact-minimum ESOPs of all r-variable Boolean functions (e. g., $\tau^1 = 1$, $\tau^2 = 2$, $\tau^3 = 3$, $\tau^4 = 6$.) Let G_i^r be an ordered set of r-variable Boolean functions whose number of product terms of exact-minimum ESOPs are i, and assume that r-variable Boolean functions are ordered in G_i^r by lexicographical order of their truth values ($\in \{0,1\}^{2^r}$.) An ordered set G^r of all r-variable Boolean functions is defined as follows :

$$G^r = G_0^r \# G_1^r \# \cdots \# G_{\tau^r}^r$$

Example 3 Table 1 shows all the 2-variable Boolean functions sorted by the minimum number of product terms of ESOPs and the lexicographical order of truth values. This table also shows the value of ν_{G^2} for every 2-variable Boolean function.

Let us consider to use $\nu_{G^r}(g_a)$ to represent g_a instead of truth values of g_a.

Def. 9 Let $y_a = \nu_{G^r}(g_a)$ ($a \in \{0,1,2\}^{n-r}$), and let $y = \langle y_a \mid a \in \{0,1,2\}^{n-r} \rangle$. $\hat{U}_{t_0}(y) \wedge \hat{R}(y)$ is called an *encoded modified reduced covering function (EMRCF)*, where $\hat{U}_{t_0}(y)$ and $\hat{R}(y)$ are defined as follows :

$$\begin{aligned} \hat{U}_{t_0}(y) &= U_{t_0}(\langle \nu_{G^r}^{-1}(y_a) \mid a \in \{0,1,2\}^{n-r} \rangle) \\ \hat{R}(y) &= R(\langle \nu_{G^r}^{-1}(y_a) \mid a \in \{0,1,2\}^{n-r} \rangle) \end{aligned}$$

To solve the equation $\hat{U}_{t_0}(y) \wedge \hat{R}(y) = 1$ exhaustively using ZBDDs, let us introduce $3^{n-r}2^r$ Boolean variables, say y_{ab} ($a \in \{0,1,2\}^{n-r}$, $0 \leq b < 2^r$), defined as the b-th digit of y_a in binary.

Clearly from the definition of ν_{G^r}, $\hat{U}_{t_0}(y)$ is a unate (negative) function of y, and the BDD size for $\hat{U}_{t_0}(y)$ is expected to be small. Fig. 2 shows the ZBDD for $\hat{U}_6$ for $n = 5$, $r = 4$. Compared with Fig. 1, structure of ZBDD is drastically simplified. Table 2 shows the ZBDD size for several cases. From this table, ZBDD size for $\hat{U}_{t_0}(y)$ is much smaller than $U_{t_0}(g)$. This fact enables us to choose $r = 3$ or $r = 4$.

Table 2: Comparison of ZBDD size for U_{t_0} and $\hat{U}_{t_0}$

n	t_0	ZBDD size for U_{t_0}: $r = 2$	$r = 4$	ZBDD size for $\hat{U}_{t_0}$: $r = 2$	$r = 4$
5	6	1,210	25,052	934	383
	7	1,425	29,554	1,107	463
	8	1,634	34,052	1,273	531
	9	1,829	38,476	1,430	575
6	6	3,856	108,002	2,986	1,721
	7	4,611	139,498	3,591	2,239
	8	5,360	170,990	4,189	2,745
	9	6,095	202,408	4,778	3,227
	10	6,824	232,802	5,360	3,682
7	6	11,794	356,852	9,142	5,735
	7	14,169	469,330	11,043	7,567
	8	16,538	581,804	12,937	9,387
	9	21,242	805,580	16,700	12,952

After the ZBDD for $\hat{U}_{t_0}(y)$ is generated, a ZBDD for $\hat{U}_{t_0}(y) \wedge \hat{R}(y)$ is generated. This step is a little more complicated than the case of conventional MRCF because of $\nu_{G^r}^{-1}$. For example, from table 1, $\nu_{G^2}^{-1}$ is derived as $g_{a11} = y_{a3}\bar{y}_{a1} \vee y_{a2}y_{a1}$, $g_{a10} = y_{a3} \oplus (y_{a2}\bar{y}_{a1})$, $g_{a01} = y_{a3}y_{a0} \vee (\bar{y}_{a2} \vee y_{a0})y_{a1}$, $g_{a00} = y_{a3}y_{a2}\bar{y}_{a0} \vee (\bar{y}_{a2} \vee \bar{y}_{a1})y_{a0}$. Size of ZBDDs which represent g_{ab} by variables y_a is smaller for greater b. From this fact, it is efficient to generate EMRCF by evaluating g_{ab} of greater b prior.

After ZBDD for $\hat{U}_{t_0}(y) \wedge \hat{R}(y)$ is generated, a solution, say y, is obtained by finding a path from the root node to the '1' leaf node of the graph, and g is decoded from y by $\nu_{G^r}^{-1}$. To make sure that it is the minimum solution, generate a ZBDD for $(\hat{U}_{t_0}(y) \wedge \hat{R}(y)) \wedge \hat{U}_{t_0-1}(y)$ and check that it is false, i. e., there is no ESOP of product terms less than $t_0 - 1$. In fact, ZBDDs for $\hat{U}_i(y)$ ($0 < i < t_0$) are easily obtained as byproducts of $\hat{U}_{t_0}(y)$.

3.2 Efficient Implementation

Using a transformation $\psi_1 \wedge \psi_2 = \psi_1 \wedge (1 \oplus \psi_1 \oplus \psi_2)$ repeatedly to the right side of Equation (2), we have :

$$R(g) = \bigwedge_{q \in \{0,2\}^{n-r}} \bigwedge_{b \in \{0,1\}^r} \left(\overline{f(q,b)} \oplus \sum_{a \in Q_q}^{\oplus} g_a(b) \right), \quad (4)$$

where $Q_q = \{a \in \{0,1,2\}^{n-r} \mid r+1 \leq \forall i \leq n.((q_i = 2 \Rightarrow a_i \neq 2) \wedge (a_i = 2 \vee a_i = q_i))\}$, and $f(\ldots,2,\ldots)$ denotes $f(\ldots,0,\ldots) \oplus f(\ldots,1,\ldots)$.

Intuitively, Q_q represents a set of cubes that cover exactly one combination of values that is also covered by a cube represented by q. This transformation is a generalization of a transformation which is originally developed for exact-minimization of generalized Reed-Muller expressions [6]

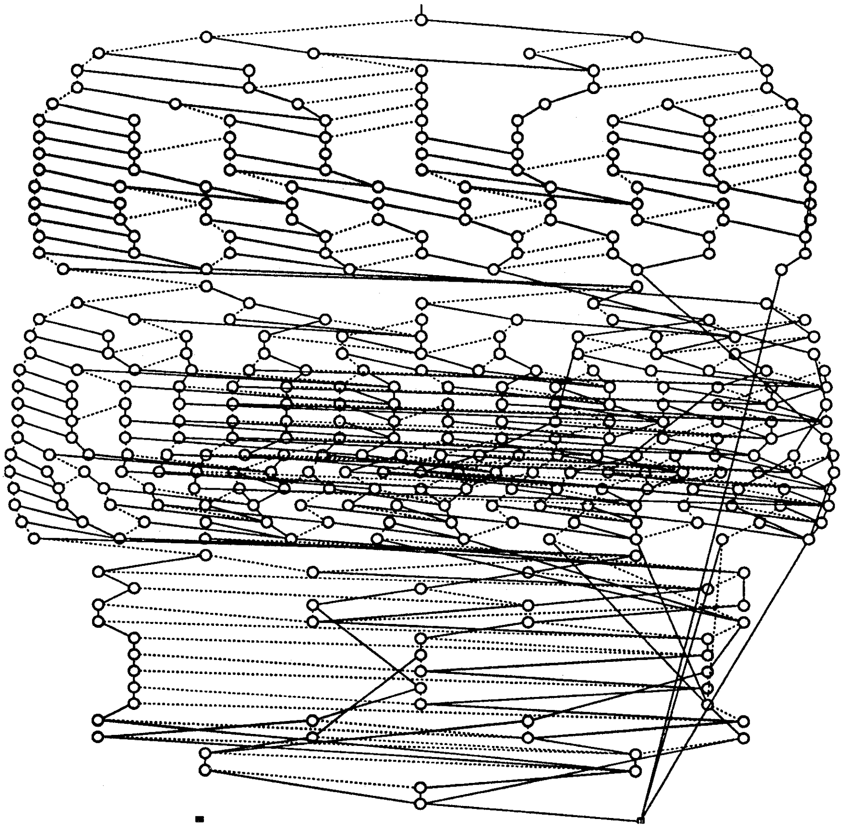

Figure 2: Encoded U_6 for $n = 5$, $r = 4$ (383 nodes)

Example 4 When $n = 3$ and $r = 1$, $R(g)$ is as follows :

$$\begin{aligned} R(g) &= \left(\overline{f(0,0,0)} \oplus g_{00,0} \oplus g_{02,0} \oplus g_{20,0} \oplus g_{22,0}\right) \\ &\wedge \left(\overline{f(0,0,1)} \oplus g_{00,1} \oplus g_{02,1} \oplus g_{20,1} \oplus g_{22,1}\right) \\ &\wedge \left(\overline{f(0,2,0)} \oplus g_{00,0} \oplus g_{01,0} \oplus g_{20,0} \oplus g_{21,0}\right) \\ &\wedge \left(\overline{f(0,2,1)} \oplus g_{00,1} \oplus g_{01,1} \oplus g_{20,1} \oplus g_{21,1}\right) \\ &\wedge \left(\overline{f(2,0,0)} \oplus g_{00,0} \oplus g_{02,0} \oplus g_{10,0} \oplus g_{12,0}\right) \\ &\wedge \left(\overline{f(2,0,1)} \oplus g_{00,1} \oplus g_{02,1} \oplus g_{10,1} \oplus g_{12,1}\right) \\ &\wedge \left(\overline{f(2,2,0)} \oplus g_{00,0} \oplus g_{01,0} \oplus g_{10,0} \oplus g_{11,0}\right) \\ &\wedge \left(\overline{f(2,2,1)} \oplus g_{00,1} \oplus g_{01,1} \oplus g_{10,1} \oplus g_{11,1}\right) \end{aligned}$$

Using $\hat{R}(y)$ of the form based on Equation (4) and employing the variable ordering of ZBDD illustrated in the following example, peak size of ZBDDs is reduced.

Example 5 The employed variable ordering for $n = 3$, $r = 1$ is as follows :

$$\begin{aligned} & y_{22,1} \succ y_{22,0} \succ \\ & y_{21,1} \succ y_{21,0} \succ y_{20,1} \succ y_{20,0} \succ \\ & y_{12,1} \succ y_{12,0} \succ y_{02,1} \succ y_{02,0} \succ \\ & y_{11,1} \succ y_{11,0} \succ y_{10,1} \succ y_{10,0} \succ \\ & y_{01,1} \succ y_{01,0} \succ y_{00,1} \succ y_{00,0} \end{aligned}$$

4 Implementation and Evaluation

4.1 Implementation

A program based on the proposed algorithm is developed and evaluated on Sun SPARC station 10 with 192MB main memory. The employed 0-suppressed BDD package requires about 25 bytes per a node. For comparison, another version based on the conventional MRCF method is also implemented. Variable ordering for the latter version is the same as [3].

Both EMRCF version and MRCF version program include a preprocessor to obtain a quasi-minimum ESOP for the given Boolean function to set the parameter t_0 for the main routine. This preprocessor is based on EXMIN2 [1].

4.2 Experimental Results

Table 3 shows the time and space required for generating exact-minimum ESOP for several randomly generated Boolean functions. For all these sample Boolean functions, the preprocessor generated exact-minimum solutions, therefore, this table represents the required resources for the preprocessor plus those for the main routine which guaranteed that there is

Table 3: Experimental results

Function		EMRCF			MRCF			B/A
n	t	r	#node(Peak) A	time (sec.)	r	#node(Peak) B	time (sec.)	
5	6	3	6,538	1.33	2	9,684	1.81	1.48
	7	3	19,818	2.76	2	31,372	3.50	1.58
	8	4	32,790	9.64	3	93,160	13.67	2.84
	9	4	81,821	17.69	4	153,746	20.79	1.88
6	6	3	20,378	3.92	2	37,294	6.61	1.83
	7	3	55,858	10.71	2	112,742	16.43	2.02
	8	4	239,328	90.50	2	408,044	83.37	1.70
	9	4	927,881	258.29	1	1,713,586	362.65	1.85
	10	4	3,562,054	1,037.82	2	6,565,711	1,528.84	1.84
7	6	3	45,345	14.66	1	86,715	27.34	1.91
	7	4	149,600	190.41	1	419,499	99.89	2.80
	8	4	568,133	328.79	2	1,184,208	242.68	2.08
	9	4	2,695,316	828.31	2	4,954,085	964.17	1.84
8	6	3	187,284	160.08	1	257,173	264.41	1.37
	7	4	415,605	1,431.46	1	872,179	473.56	2.10
	8	3	2,155,435	1,069.54	3	5,985,686	2,084.39	2.78
9	6	4	726,147	10,569.13	1	942,635	2,606.26	1.30
	7	4	1,067,834	10,382.22	2	3,373,383	2,586.07	3.16
	8	3	6,431,434	5,465.67	*	(> 8M)	—	—

n : Number of input variables of the given Boolean function
t : Number of product terms of an exact-minimum ESOP for the given Boolean function
EMRCF, MRCF : Results on programs based on proposed EMRCF and conventional MRCF, respectively (best r is chosen among 1, 2, 3, and 4 to reduce #node(Peak))
#node(Peak) : Maximum number of nodes required during computation
time : Elapsed (real) time for computation on Sun SPARC station 10 with 192 MB main memory

not any better solution than one obtained by the preprocessor. For each samples, this table shows the result among 1, 2, 3, and 4 for the parameter r that require smallest space for ZBDD.

From this table, a 7-variable Boolean function whose minimum ESOP is 7 terms is minimized within 149,600 nodes of ZBDD space by EMRCF version, while 419,499 nodes of ZBDD space are required by MRCF version. In case of a sample Boolean functions of (n,t)=(9,7), required space is reduced to 1/3.16. Within 192 MB main memory, 6-, 7-, 8-, and 9-variable Boolean functions of up to respectively 10, 9, 8 and 8 product terms in ESOP are minimized, and in all cases, required space for the EMRCF version is smaller than the MRCF version.

From these experiments, $r = 4$ seems an optimal choice to reduce space, and $r = 3$ seems an optimal choice to reduce time for the EMRCF version in most cases. For example, sample Boolean functions of (n,t)=(9,6) and (9,7) are minimized in 1,272.39 sec. and 1,213.53 sec., respectively, by choosing $r = 3$, although requiring 843,748 nodes and 6,431,434 nodes, respectively, of ZBDD space are required. While $r = 2$ or $r = 1$ seems an optimal choice for the MRCF version.

5 Conclusion

In this paper, we have discussed on the exact-minimization of ESOP for a given Boolean function, and introducing a novel encoding method to reduce the size of BDDs, a new algorithm based on an EMRCF, an improvement of the MRCF [3], is proposed.

From experimental result, a 7-variable Boolean function whose minimum ESOP is 7 terms is minimized by the proposed method within 149,600 nodes of ZBDD space, while 419,499 nodes of ZBDD space is required by the conventional MRCF-based method. Within 192 MB main memory, 6-, 7-, 8-, and 9-variable Boolean functions of up to respectively 10, 9, 8 and 8 product terms in ESOP are minimized, and in all cases, required space is reduced compared with the conventional MRCF-based method.

The developed method is expected to be used for EXOR-based design, e. g., technology mapping and FPGA design, and evaluation of the quasi-minimization algorithms for deriving ESOPs.

References

[1] T. Sasao : "EXMIN2: A simplification algorithm for exclusive-OR-Sum-of-products expressions for multiple-valued input two-valued output functions," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 12, no. 5, pp. 621–632, May 1993.

[2] 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.

[3] T. Sasao : "An exact minimization of AND-EXOR expressions using reduced covering functions," Proc. SASIMI '93, pp. 374–383, Aug. 1993.

[4] R. E. Bryant : "Graph-based algorithms for Boolean function manipulation," IEEE Trans. Comput., vol. 35, no. 8, pp. 677–691, Aug. 1986.

[5] S. Minato : "Zero-suppressed BDDs for set manipulation in combinatorial problems", Proc. 30th ACM/IEEE DAC, pp. 272–277, June 1993.

[6] T. Sasao and D. Debnath : "An exact minimization algorithm for generalized Reed-Muller expressions", Proc. IEEE APCCAS '94, pp. 460–465, Dec. 1994.