設計自動化 68－8
(1993. 10. 28)

BDDを用いたAND-EXOR論理式最小化の一手法

笹尾 勤, 松浦宗寛
九州工業大学 情報工学部 電子情報工学科
〒820 飯塚市大字川津680-4

あらまし: 任意の積項をEXORで結合した論理式をAND-EXOR論理式(ESOP)という. 本論文は, ESOPが与えられた関数を表現するための条件を$3^{(n-r)} \cdot 2^r (r = 0, 1, 2, 3, 4)$変数の修正規約被覆関数$R(g)$を用いて定式化し, 最小ESOPが$R(g) = 1$を満たす割当の内で, gの重みが最小のもの対応することを示す. また, 最小の割当をBDDを用いて求める方法を示す. 最後に, 本手法を, $n = 9$までの論理関数のESOPに適用した場合の実験結果を示す. また, 多出力関数への拡張についても触れる.

和文キーワード　EXOR, ESOP, Reed-Muller論理式, 論理式最小化, BDD

A Method to Derive Exact Minimimum AND-EXOR Expressions using Binary Decision Diagrams

Tsutomu SASAO and Munehiro MATSUURA
Department of Computer Science and Electronics
Kyushu Institute of Technology
680-4 Kawazu Iizuka, 820, Japan

Abstract: This paper presents a method to derive an EXOR sum-of-products expression (ESOP) having the minimum number of products for a given logic function. The minimization method uses a modified reduced covering function, which is an improvement of the method proposed by Perkowski and Chrzanowska-Jeske. Zero-supressed binary decision diagrams (BDDs) are used to obtain exact solutions. Various techniques to reduce computation time and memory storage are developed. Experimental results up to 9 variables are shown. An extension to multiple-output functions is also shown.

英文 key words　EXOR, ESOP, Reed-Muller expressione, logic minimization BDD.

表 1: 算術演算回路を表現する論理式の比較

Data	積項数		リテラル数	
	SOP	ESOP	SOP	ESOP
ADR4	75	31	423	168
LOG8	123	96	1019	785
MLP4	121	61	889	441
NRM4	120	73	887	602
RDM8	76	31	406	181
ROT8	57	35	389	280
SQR8	180	114	1398	809
WGT8	255	54	2078	356

表 2: n 変数乱数関数を表現する論理式の積項数の比較

n	$\|f\|$	ESOP	SOP
4	8	3	4
5	16	6	6
6	32	10	13
7	64	19	24
8	128	40	46
9	256	76	86
10	512	149	167
11	1024	289	331
12	2048	576	611

1 はじめに

現在, 論理回路は, AND, OR, 及び NOT を基本とした設計理論に基づいて設計しており, 制御回路等では高品質の回路が得られる. しかし, 算術演算回路, 誤り訂正回路, 通信用回路などでは, EXOR ゲートを巧妙に使うと回路のゲート数を大幅に削減できる.

任意の積項を EXOR で結合した論理式を AND-EXOR 論理式 (ESOP) という. ESOP は AND-EXOR 形論理式中, 最も一般的な論理式である [24, 25].

必要な積項数を論理和形 (SOP) と比較してみると, 多くの場合, ESOP の方が SOP よりも少ない (表 1 ~ 4). SOP と ESOP に関して, 任意の n 変数関数を実現するために十分な積項数は, SOP では 2^{n-2}, ESOP では 2^{n-2}である. また, 任意の $n(=2r)$ 変数対称関数は, 積項数が高々$2^r \cdot 3$ の ESOP で実現できる. 対称関数を実現する最小 ESOP の積項数は, SOP の積項数を越えない』ことが $n \leq 7$ の場合に確かめられ, $n \geq 8$ の場合にも成立すると推論されていた [23]. 最近, この推論が任意の n で成立することが証明された [21]. 対称関数は, 計数の基本演算であり, 計数はあらゆる算術演算の基本となる [5]. 従って, 算術演算回路の設計には, EXOR が有効であることが, 理論的にも裏付けられたと言える.

ESOP は $n \leq 5$ の場合は, 網羅的 (に近い) 方法で最適解が求められている [11, 12, 13]. 従って, $n \leq 5$ の場合の ESOP の最適化問題は, 解決されたといえる. Perkowski らは, ESOP が与えられた関数を表現するための条件を 3^n変数の論理関数 $H(g)$(Helliwell 関数) で表現し, 最小 ESOP が $H(g)=1$ を満たす割当の内で, g の重みが最小のものに対応することを示した. しかし, この方法を率直にプログラムしたものは $n=4$ までしか実用にならなかった. 従って, $n \geq 6$ の場合は専ら繰り返し改善法が用いられていた [24, 26, 1].

本論文では, ESOP が与えられた関数を表現するための条件を $3^{(n-r)} \cdot 2^r (r = 0,1,2,3,4)$ 変数の修正規約被覆関数 $R(g)$ を用いて定式化し, 最小 ESOP が $R(g)=1$ を満たす割当の内で, g の重みが最小のものに対応することを示す. また, この問題の解を BDD を用いて能率的に求める方法を示す. また, 実験により本手法が, $n=9$ までの論理関数の ESOP に適用可能なことを示す.

表 3: 4 変数論理式の積項数の分布

t	ESOP	SOP
0	1	1
1	81	81
2	2268	1804
3	21744	13472
4	37530	28904
5	3888	17032
6	24	3704
7		512
8		26
t	3.66	4.13

表 4: 各種関数を実現するための積項数 ($n=2r$)

	SOP	ESOP
任意関数	2^{n-1}	2^{n-2}
対称関数	2^{n-1}	$2 \cdot 3$
パリティ関数	2^{n-1}	n
$x_1x_2 \vee x_3x_4 \cdots \vee x_{n-1}x_n$	r	2^r-1
4 変数関数	4.13	3.66

2 最小化問題の定式化

fを n 変数関数 ($n \geq 5$), $B=\{0,1\}$, $T=\{0,1,2\}$ とする. 論理変数のベクトルを $X=(x_1,x_2,\cdots,x_n)$ とする. $X=(X_1,X_2)$ を X の分割とし, X_1の変数は $(n-r)$ 個, X_2の変数は r個, $r=0,1,2,3,4$ とする. このとき, fは

$$f(X_1,X_2)=\sum_{\boldsymbol{a}} \oplus X_1^{\boldsymbol{a}} \cdot g(\boldsymbol{a}:X_2) \tag{1}$$

と展開できる. ここで,$\boldsymbol{a} \in T^{(n-r)}$である. また,

$$X_1^{\boldsymbol{a}}=x_1^{a_1} \cdot x_2^{a_2} \cdot \cdots \cdot x_{n-r}^{a_{n-r}}, \quad x_i^{a_i}=\begin{cases} \bar{x}_i & a_i=0 \\ x_i & a_i=1, \\ 1 & a_i=2 \end{cases}$$

であり, $g(\boldsymbol{a}:X_2)$ は r変数関数である. また, ベクトル $\boldsymbol{a}$ は 3^{n-r}個の値をとる. (1) の X_1に $\boldsymbol{b} \in B^{n-r}$を代入することにより,

$$f(\boldsymbol{b},X_2)=\sum_{\boldsymbol{c} \geq_R \boldsymbol{b}} \oplus g(\boldsymbol{c}:X_2) \tag{2}$$

を得る. ここで, $\sum \oplus_{\boldsymbol{c} \geq_R \boldsymbol{b}}$は $\boldsymbol{c} \geq_R \boldsymbol{b}$ を満たすベクトル $\boldsymbol{c}$に関する排他的論理和であり, $\geq_R$は 2 項関係 $\{(0,0),(1,1),(2,2),(2,0),(2,1)\}$ を表す. $\boldsymbol{c} \geq_R \boldsymbol{b}$ を満たすベクトルは, 一つの $\boldsymbol{b}$ に対して, 2^{n-r}個存在する. $\boldsymbol{b}$ の定め方は 2^{n-r}個存在するので, (2) も 2^{n-r}個存在する.

次に (2) の X_2に, B上の r項ベクトル $\boldsymbol{d}=(d_{n-r},d_{n-r+1},\cdots,d_n)$ を代入することにより,

$$f(\boldsymbol{b},\boldsymbol{d})=\sum_{\boldsymbol{c} \geq_R \boldsymbol{b}} \oplus g(\boldsymbol{c}:\boldsymbol{d}) \tag{3}$$

を得る. (3) は $\boldsymbol{b}$が2^{n-r}個, $\boldsymbol{d}$が2^r存在するので, 全部で$2^{n-r} \cdot 2^r = 2^r$個存在する. ここで $g(\boldsymbol{c}:\boldsymbol{d})$ は 0 または 1 である. (3) が全ての $\boldsymbol{b}$ と $\boldsymbol{d}$に関して同時に成立するための必要十分条件は

$$\bigwedge_{(\boldsymbol{b},\boldsymbol{d})} ([\sum_{\boldsymbol{c} \geq_R \boldsymbol{b}} g(\boldsymbol{c}:\boldsymbol{d})] \oplus f(\boldsymbol{b},\boldsymbol{d}) \oplus 1)=1 \tag{4}$$

ここで$\bigwedge_{(\boldsymbol{b},\boldsymbol{d})}$は $\boldsymbol{b} \in B^{n-r}$と $\boldsymbol{d} \in B^r$を満たす全ての組合せに関する論理積である. $g(\boldsymbol{c}:\boldsymbol{d})$はブール変数で, $f(\boldsymbol{b},\boldsymbol{d})$ の値は与えられた関数から定まる. (4) を規約被覆関数 (RCF) という. RCF の変数は全部で$2^r \cdot 3^{n-r}$個存在する. fの最小 ESOP の一般形は (1) の形で表現できる. $g(\boldsymbol{a}:X_2)$ を未知の r変数論理関数とし, 3^{n-r}個の論理関数 $g(\boldsymbol{a}:X_2)$ を求める. 各関数 $g(\boldsymbol{a}:X_2)$ は, $g(\boldsymbol{c}:\boldsymbol{d})$ の値により定まる.

この際, RCF を満たすよ うに $g(\boldsymbol{c}:\boldsymbol{d})$ を定める. RCF を満たす解のうち, 次に示すコス トが最小のものが, 最小 ESOP に対応する.

$$\sum_{\boldsymbol{a}} \tau(g(\boldsymbol{a}:X_2)) \tag{5}$$

但し$\sum$は全ての $\boldsymbol{a} \in T^{n-r}$に関する算術和である. 今までに, 4 変数までの最小 ESOP の表が得られているので, $r=0,\cdots,4$ の場合, $\tau(g(\boldsymbol{a}:X_2))$ の値は既知である. 従って次の定理が成立する.

定理 1 n 変数関数の最小 ESOP は, $2^r\cdot 3^{n-r}$変数 $(r=0,\cdots,4)$ の RCF を満たす割当の うちで, (5) の値が最小のものに対応する.

コス ト関数について.

1) $r=0$ のとき. $g(\boldsymbol{a}:X_2)=g(\boldsymbol{a})$ は 0 変数関数で X_2には依存しない. $g(\boldsymbol{a})$ は全部で 3^n存在し, それぞれが n 変数の積項 (全部で 3^n個存在する). RCF は Helliwell 関数 [20] と一致する. コス ト関数は

$$\tau(g(\boldsymbol{a}:X_2))=g(\boldsymbol{a}) \ (0 \text{ または } 1)$$

となる.

2) $r=1$ のとき. $g(\boldsymbol{a}:X_2)$ は 1 変数関数で, $g(\boldsymbol{a}:X_2)=\overline{X_2}\cdot g(\boldsymbol{a}:0)\vee X_2\cdot g(\boldsymbol{a}:1)$ と展開できる. 1 変数関数を表現するには積項数が高々1 個あれば十分であ り, $g(\boldsymbol{a}:0)\vee g(\boldsymbol{a}:1)=1$ となるときのみ積項が必要である. 従ってコス ト関数は

$$\tau(g(\boldsymbol{a}:X_2))=g(\boldsymbol{a}:0)\vee g(\boldsymbol{a}:1)$$

となる.

3) $r=2$ のとき, $g(\boldsymbol{a}:X_2)=g(u,v)$ は 2 変数関数である. 2 変数関数をESOP で表現するには積項数は高々2 個あれば十分であり, コス ト関数は

$$\tau(g(\boldsymbol{a}:X_2))=\begin{cases} 0 & g(u,v)=0 \text{ のとき.} \\ 1 & g(u,v)=1,\bar{u},u,\bar{v},v,\bar{u}\cdot\bar{v}, \\ & \bar{u}\cdot v,u\cdot\bar{v},u\cdot v \text{ のとき.} \\ 2 & \text{上以外のとき.} \end{cases}$$

となる.

4) $r=3$ のとき, $g(\boldsymbol{a}:X_2)$ は 3 変数関数で, 積項数が高々3 の ESOP で表現できる. コス ト関数は複雑になるが, 論理式で表現可能である.

5) $r=4$ のとき, $g(\boldsymbol{a}:X_2)$ は 4 変数関数で, 積項数が高々6 の ESOP で表現できる. コス ト関数は複雑になるが, 論理式で表現可能である.

3 例題

3 変数関数 $f(x,y,z)$ において, $X=(x,y,z)$, $X_1=(x,y)$, $X_2=(z)$ と分割する. fを x と yで展開することによ り,

$$\begin{aligned} f(x,y,z)=&\bar{x}\cdot\bar{y}\cdot g(0,0:z)\oplus\bar{x}\cdot y\cdot g(0,1:z)\oplus\bar{x}\cdot 1\cdot g(0,2:z)\oplus \\ &x\cdot\bar{y}\cdot g(1,0:z)\oplus x\cdot y\cdot g(1,1:z)\oplus x\cdot 1\cdot g(1,2:z)\oplus \\ &1\cdot\bar{y}\cdot g(2,0:z)\oplus 1\cdot y\cdot g(2,1:z)\oplus 1\cdot 1\cdot g(2,2:z) \end{aligned} \tag{6}$$

このとき, $g(i,j:z)$ は, 1 変数関数である. x と yに 0,1 を代入することによ り次式を得る.

$$f(0,0,z)=g(0,0:z)\oplus g(0,2:z)\oplus g(2,0:z)\oplus g(2,2:z) \tag{7}$$
$$f(0,1,z)=g(0,1:z)\oplus g(0,2:z)\oplus g(2,1:z)\oplus g(2,2:z) \tag{8}$$
$$f(1,0,z)=g(1,0:z)\oplus g(1,2:z)\oplus g(2,0:z)\oplus g(2,2:z) \tag{9}$$
$$f(1,1,z)=g(1,1:z)\oplus g(1,2:z)\oplus g(2,1:z)\oplus g(2,2:z) \tag{10}$$

(7) よ り,

$$f(0,0,0)=g(0,0:0)\oplus g(0,2:0)\oplus g(2,0:0)\oplus g(2,2:0)$$
$$f(0,0,1)=g(0,0:1)\oplus g(0,2:1)\oplus g(2,0:1)\oplus g(2,2:1)$$

(8) よ り,

$$f(0,1,0)=g(0,1:0)\oplus g(0,2:0)\oplus g(2,1:0)\oplus g(2,2:0)$$
$$f(0,1,1)=g(0,1:1)\oplus g(0,2:1)\oplus g(2,1:1)\oplus g(2,2:1)$$

(9) よ り,

$$f(1,0,0)=g(1,0:0)\oplus g(1,2:0)\oplus g(2,0:0)\oplus g(2,2:0)$$
$$f(1,0,1)=g(1,0:1)\oplus g(1,2:1)\oplus g(2,0:1)\oplus g(2,2:1)$$

(10) よ り,

$$f(1,1,0)=g(1,1:0)\oplus g(1,2:0)\oplus g(2,1:0)\oplus g(2,2:0)$$
$$f(1,1,1)=g(1,1:1)\oplus g(1,2:1)\oplus g(2,1:1)\oplus g(2,2:1)$$

以上の式が同時に成立するための必要十分条件は, $R(\boldsymbol{g})=1$, ここで,

$$\begin{aligned} R(\boldsymbol{g})=&(g(0,0:0)\oplus g(0,2:0)\oplus g(2,0:0)\oplus g(2,2:0)\oplus\overline{f(0,0,0)})\cdot \\ &(g(0,0:1)\oplus g(0,2:1)\oplus g(2,0:1)\oplus g(2,2:1)\oplus\overline{f(0,0,1)})\cdot \\ &(g(0,1:0)\oplus g(0,2:0)\oplus g(2,1:0)\oplus g(2,2:0)\oplus\overline{f(0,1,0)})\cdot \\ &(g(0,1:1)\oplus g(0,2:1)\oplus g(2,1:1)\oplus g(2,2:1)\oplus\overline{f(0,1,1)})\cdot \\ &(g(1,0:0)\oplus g(1,2:0)\oplus g(2,0:0)\oplus g(2,2:0)\oplus\overline{f(1,0,0)})\cdot \\ &(g(1,0:1)\oplus g(1,2:1)\oplus g(2,0:1)\oplus g(2,2:1)\oplus\overline{f(1,0,1)})\cdot \\ &(g(1,1:0)\oplus g(1,2:0)\oplus g(2,1:0)\oplus g(2,2:0)\oplus\overline{f(1,1,0)})\cdot \\ &(g(1,1:1)\oplus g(1,2:1)\oplus g(2,1:1)\oplus g(2,2:1)\oplus\overline{f(1,1,1)}). \end{aligned}$$

コス ト関数は,

$$\sum_{(i,j)}(g(i,j:0)\vee g(i,j:1)) \tag{11}$$

である. ここで$\sum_{(i,j)}$は $(i,j)\in T^2$に関する算術和となる.

4 BDD を用いた解法

与えられた関数の最小 ESOP を求めるには, RCF を満たす割当のうち, コス トが最小の割当を求めればよい. ここでは, BDD(Binary Dicision Diagram)[4, 16] を用いた解法について述べる.

組合せ問題の制約条件を論理方程式 $H(\boldsymbol{g})=1$ で表現すると. $H(\boldsymbol{g})=1$ を満たす割当が許容解となる. 論理関数 $H(\boldsymbol{g})$ を BDD で表現すると, BDD の根から定数 1 までの経路 (1 パス) が許容解に対応する. また, 経路にコス トを適当に付加することによ り, 各経路が解のコス トを表現するよ うにできる. 従って, 組合せ最適化問題は, コス ト最小の経路を求めることに対応する [15].

5 RCF の変数の個数

与えられた論理関数の変数の個数を n とし, r変数の最小 ESOP の表を使用する場合, RCF の変数の個数は$\xi(n,r)=2^r\cdot 3^{n-r}$となる. 表 5に必要な変数の個数を示す. rが大きいほど, 変数の個数は少な くなるが, コス ト関数や関数 $U(t_1)$ の計算に時間がかかるよ うになる.

6 計算時間と メモ リの削減法

BDD を用いた問題の定式化は簡単であるが, BDD の生成には大量の メモ リが必要となる. また, 計算時間の殆どが, BDD 生成のための時間であ り, 最短経路を見つ けるための時間は, 比較的短い. 従って, 如何にして能率よ くBDD を生成するかが問題となる.

表 5: RCF の変数の個数

n	$\xi(n,r)$				
	$r=0$	$r=1$	$r=2$	$r=3$	$r=4$
4	81	54	36	24	16
5	243	162	108	72	48
6	729	486	324	216	144
7	2187	1458	972	648	432
8	6551	4373	2916	1944	1296
9	19653	13119	8748	5832	3888

6.1 ESOP の積項数の下界

BDD を能率良く生成するには種々の方法があるが, 最良の方法は, BDD を生成しないことである. ESOP 簡単化プログラム EXMIN2 は性能が良く, 多くの場合に最適解を生成する. 関数 f を表現する ESOP を EXMIN2 で簡単化した結果の積項数を t_1 としよう. また, f を表現する ESOP の積項数の下界を t_2 とする. もし, $t_1 = t_2$ ならば, EXMIN2 で求めた ESOP は最小であり, BDD を生成せずに, アルゴリズムを停止できる. 次の定理は関数 f の ESOP の積項数の下界を与える.

定理 2 関数 f が $f = \bar{x}_i \cdot f_{i_0} \oplus x_i \cdot f_{i_1}$ と展開できるとき, f の ESOP の積項数は少なくとも $\max\{L_1, L_2\}$ である. ここで,

$$L_1 = \frac{1}{2}\max\{\tau(f_{i_0}), \tau(f_{i_1}), \tau(f_{i_2})\}$$
$$L_2 = 1 + \min_{q_i}\{L_1(q_i \oplus f)\}$$

である. また, q_i は f の最小項を少なくとも一つ含む積項, $\tau(g)$ は g の最小 ESOP の積項数を示す.

上の定理を用いるには, $(n-1)$ 変数関数の最小 ESOP の積項数を求める必要があるが, これは $n = 5$ までは, 索表により実行 [13] できる.

6.2 ESOP の積項数の上界

$R(g)$ の BDD をそのまま生成するとメモリオーバーフローする場合が多いので, EXMIN2 で, 準最適解を求め, それよりも積項数の少ない解のみを調べる. EXMIN2 で求めた準最適解の積項数を t_1 とする. 積項数が t_1 より小さいという条件を表現する論理関数を $U(t_1)$ とすると,

$$U(t_1) = \begin{cases} 1 & (\sum \tau(g(\boldsymbol{a} : X_2)) < t_1 \\ 0 & \text{上以外のとき} \end{cases}$$

となる. $R(g)$ のかわりに, $U(t_1) \cdot R(g)$ の BDD を生成することによって, 必要なメモリと計算時間を大幅に削減できる. $U(t_1) \cdot R(g)$ を修正規約被覆関数 (MRCF:Modified Reduced Covering Function) という.

例 1 任意の論理関数は, 積項数が高々3 の ESOP で表現できる. 積項数が 3 より少ない ESOP の解の集合を表す論理関数を U(3) とすると,

$$U(3) = \vee_{(i,j,s,t,k,w)}(g(i,j:0) \vee g(i,j:1)) \cdot (g(s,t:0) \vee g(s,t:1)) \cdot (g(k,w:0) \vee g(k,w:1)) \quad (12)$$

ただし, $\vee$ は $(i,j) \neq (s,t)$, $(s,t) \neq (k,w)$, $(k,w) \neq (i,j)$, $i,j,s,t,k,w \in \{0,1\}$ に関する和である. 従って, 積項数が 2 以下の最小解を求めるには, $R(g) \cdot U(3) = 1$ を満たす割当の中で, (12) の値を最小にするものを求めればよい.

6.3 零サプレス BDD

MRCF は, RCF を満たす二値ベクトルの内, 重みが $t_1 - 1$ 以下のものの集合を表現する. MRCF の変数の個数は $O(3^n)$ であるが, $t_1 \leq 2^{(n-1)}$ なので, ベクトル中の非零要素数は高々 $2^{(n-1)}$ である. 従って, MRCF は重みが小さなベクトルの集合を表現しているといえる. 零サプレス BDD はこのような集合の表現に適した BDD であり[17], 通常の BDD よりも計算時間とメモリを大幅に削減できる.

6.4 乗算方法

MRCF は 2^n 個のパリティ関数の論理積で表現できる. BDD を構成する際, 一つのパリティ関数を乗ずる毎に, BDD の節点数がおよそ 2 倍ずつ増加していく. 従って, BDD の作成中にメモリがオーバーフローし, 計算が不可能となることが多い. 節点数の増加を抑えるため, なるべく変数の総数が増えないようなパリティ関数同士の積から計算していく. 乗算方法は次に示す三つの方法を試みた.

a. 最小項に対応するパリティ関数同士を乗算する場合, 各々の関数に共通の積項が多い方が, BDD で使用する変数が少なくなり, 節点数の増加は少ない. そのために, 最小項の番号をベクトルで表現したときベクトル間のハミング距離が近い順序で計算する. 各最小項に対応するパリティ関数を $g_0, \cdots, g_m$ (添字は最小項の番号, $m = 2^n - 1$) とすると,

$$g_m \cdot g_{m-\frac{m+1}{2}} \cdot g_{m-\frac{m+1}{4}} \cdot g_{m-\frac{3(m+1)}{4}} \cdots\cdots g_{\frac{3(m+1)}{4}} \cdot g_{\frac{m+1}{4}} \cdot g_{\frac{m+1}{2}} \cdot g_0$$

という順序で AND 演算を行った.

b. 実験によって, $f(\boldsymbol{a}) = 1$ となる最小項に対応するパリティ関数を先に計算すると, BDD の収束が早くなることがわかったので, a で決定した順序を前半部分と後半部分とに 2 分割し, $f(\boldsymbol{a}) = 1$ となる最小項に対応するパリティ関数の多い方を優先する.

c. b の方法をさらに徹底して行う. 方法 b によって優先順位の決定した 2 つの部分について, 優先順位の高い方については, 方法 b と同様に 2 分割し優先順位を決定する. 優先順位の低い方は, 2 分割しその前半部分を無条件に優先する. この操作を分割できなくなるまで繰り返す.

7 最小化アルゴリズム

ステップ 1. $F \leftarrow$ EXMIN2 で簡単化した結果.
$t_1 \leftarrow F$ の積項数.
$t_2 \leftarrow$ 定理 2で求めた下界.
ステップ 2. $t_1 = t_2$ ならば, F は最小なので停止.
ステップ 3. $R(g) \cdot U(t_1)$ の BDD を生成する.
BDD が関数 0 を表現する場合には, F は最小なので停止.
ステップ 4. BDD の最短経路を求める.
ステップ 5. 最短経路に対応する ESOP を求める.

8 実験結果

与えられた論理式から, RCF の BDD を生成し, 最小 ESOP を求めるプログラムを C 言語で開発した. ただし, BDD パッケージとして, [2] を基にしたものと [17] を基にしたものを二種類開発した. また, EXMIN2 と積項数の下界を求めるプログラムは別のパッケージであり, 別々に実行した.

$n = 4$ の場合は, 網羅的方法で, 全ての論理関数の最適解が求まっている. $n = 5$ の場合は, 実験した関数はすべて, 実用時間内で最適解が求まった. $n = 5 \sim 9$ までの実験結果を表 6, 7に示す. この表は, 各 n に対して, 計算時間が最小になった場合の r の値とその際の, 最大使用節点数, 最終 BDD の節点数, 計算時間を示してある. 表 6 は, 従来の BDD を使用した場合, 表 7は, 零サプレス BDD を使用した場合のデータを示している. 計算時間はステップ3~5 の分を示している. 零サプレス BDD を用いると, 従来の BDD ではメモリオーバーフローのために解けなかった問題が解けている. 使用計算機は HP9000/720(HP720:64MB メインメモリ) である.

次に, r の選び方, 乗算方法, BDD のデータ構造による影響を調べるために 5 変数関数で最小 ESOP の積項が 7 になる関数を最小化した結果を表 8に示す. これより, 従来の BDD よりも零サプレス BDD を用いた方が性能が上がっていることがわかる.

従来の BDD と零サプレス BDD との比較のために, 両方で解が得られたデータについて, 最大使用節点数, 最大 BDD の節点数, 最終 BDD の節点数, 計算時間の平均値について従来の BDD を 1.00 とした場合の比を表 9に示す. これより, 零サプレス BDD を使用するとメモリ使用量はおよそ 50%, 計算時間は 15% ほど削減できることがわかる.

6.4で述べた三つの乗算方法 a, b, c で BDD の生成を行った. すべての方法で解が得られたデータについて, 最大使用節点数, 最大 BDD の節点数, 計算時間の平均値について, 従来の BDD で方法 a を用いた場合を 1.00 とした場合の比を表 10に示す. 方法 c は, 真理値表濃度が低い時に効果があることが予測できる. 実験でも, 6 変数以上の実験に用いた関数は真理値表濃度が低く, かなり良い結果となっている.

多出力関数の最小化については, mlp2 の最小解を BDD から求めることができた. また, adr2 については, 6 積項の解が存在しないことを証明できた. この場合の使用計算機は Sparc station 10(SS10:256MB メイン メモリ) である. 零サプレス BDD を用いた場合は, mlp2 の最小解が HP9000/720(HP720:64MB メイン メモリ) を用いて求まった.

9 まとめ

従来, ESOP の最小化は非常に困難であると考えられていた. Perkowski らは Helliwell 関数 [20] を用いた ESOP の最小化法を示しているが, これを率直にプログラムしたものは, $n = 4$ までしか使えなかった. 本論文では, 最小化問題を MRCF で定式化し, BDD を用いることによって, $n = 9$ までの論理関数の ESOP の最小化が可能であることを実証した. また, この方法を多出力関数の最小化にも拡張した.

$n \leq 5$ の場合は, 最小 ESOP を索表で求めることができるので, 本手法は, $n \geq 6$ の場合に有用できる. 最小化は, 時間がかかるので, 通常は, ヒューリスティックな簡単化アルゴリズム EXMIN2[26] を用いればよい. EXMIN2 では, 多変数多出力の ESOP を現実的な時間で簡単化できる.

EXOR ゲートを用いると, 算術演算回路や通信用回路を効率よく実現できる. また, 一般の回路も簡単になる場合もある. EXOR ゲートは OR ゲートよりも高価な場合には, 一部の EXOR を OR で置き換える手法も開発されている [24]. また, FPGA(Feild Programmable Gate Array)[3] の場合, EXOR と OR のコストは同じなので, EXOR を積極的に用いて設計した方が有利である.

謝辞

本研究は一部, 文部省科学研究費補助金による. 下界を求めるプログラムを提供頂いた徳山高専, 神田徳夫教授の深謝する. また, 湊真一氏には零サプレス BDD に関して有益なご助言を頂いた.

参考文献

[1] D. Brand and T. Sasao, "Minimization of AND-EXOR expressions using rewriting rules", *IEEE Transactions on Computers*, Vol.C-42, No.5, May 1993, pp.568-576.

[2] K. S. Brace, R. L. Rudell and R. E. Bryant, " Efficient implementationof a BDD package", *Proc. 27th Design Automation Conference*, June 1990, pp. 40-45.

表 6: MRCF を用いた ESOP の最小化 (従来の BDD)

n	t	r	# of nodes		time
			max	final	(sec)
5	5	2	14120	110	7.7
	6	3	42733	74	15.2
	7	3	102569	2025	43.7
	8	4	174804	8103	90.6
	9	4	371494	51763	163.3
6	3	2	6637	366	7.7
	4	3	25578	250	16.8
	5	2	56552	934	31.6
	6	2	211731	724	112.9
	7	2	750040	7761	391.9
7	3	3	27274	682	19.7
	4	3	126389	682	96.7
	5	3	349758	970	227.9
8	3	3	84769	1946	85.9
	4	3	704318	1946	912.2
9	3	3	244300	5834	620.0

表 7: MRCF を用いた ESOP の最小化 (零サプレス BDD)

n	t	r	# of nodes		time
			max	final	(sec)
5	5	2	9069	7	7.7
	6	2	28076	14	14.7
	7	3	88217	830	42.1
	8	4	123130	3732	92.4
	9	4	234084	21887	143.2
6	3	3	8772	11	7.4
	4	2	14538	10	14.3
	5	2	37123	25	24.8
	6	2	114080	40	77.9
	7	2	369273	354	285.5
7	3	3	28062	5	18.7
	4	3	57118	12	70.7
	5	2	173039	19	164.2
	6	2	581322	34	609.0
8	3	3	85812	5	103.5
	4	3	202363	6	599.5
	5	3	1097703	7	1441.6
9	3	3	227110	5	634.2
	4	2	631832	14	2974.2

n : 入力変数の個数
t : 最小 ESOP の積項数
r : 定理 1のパラメータ
max : 計算中に使用した最大節点数
final : MRCF の最終節点数
time : CPU time
HP 9000 Model 720 Workstation 64 MB main memory.

[3] S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic, *Field Programmable Gate Arrays*, Kluwer Academic Publishers, Boston 1992.

[4] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. Comput.* Vol. C-35, No.8, Aug. 1986, pp.677-691.

[5] C. Damm," How much EXOR improves on OR?", IFIP WG. 10. 5 Workshop on Application of the Reed-Muller expansion in Circuit Design, Sept 1993.

[6] M. Davio, J-P Deschamps, and A. Thayse, "Discrete and switchingfunctions", McGraw-Hill International, 1978.

[7] 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.

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

[9] M. Helliwell and M. Perkowski, "A fast algorithm to minimize multi-output mixed-polarity generalized Reed-Muller forms",

表 8: 5 変数 7 積項関数

乗算方法	r	従来の BDD			零サプレス BDD		
		# of nodes		time	# of nodes		time
		max	max BDD	(sec)	use	max BDD	(sec)
a	0	498940	303579	151.1	119925	68468	107.6
	1	327541	201772	91.8	114357	66517	77.6
	2	233051	138858	62.3	107980	64271	58.7
	3	162325	97664	48.7	90665	54811	46.6
	4	102569	55807	63.4	84240	45111	65.2
b	0	398577	243450	119.8	97935	55872	87.3
	1	264133	160871	74.8	91714	53743	65.8
	2	187550	114658	52.9	88675	53833	49.3
	3	137175	84337	43.7	88217	51384	42.1
	4	104273	56382	63.8	86911	45194	65.9
c	0	374972	235842	116.3	97921	55872	84.4
	1	246778	154823	71.6	91714	53743	65.2
	2	174852	110565	51.9	88675	53833	47.8
	3	142567	84334	44.0	88217	51384	42.3
	4	104273	65233	67.5	87015	57713	69.2

r : 定理 1のパラメータ
max : 計算中に使用した最大節点数
max BDD : MRCF の BDD 作成中の最大 BDD の節点数
time : CPU time
HP 9000 Model 720 Workstation 64 MB main memory.

表 9: 従来の BDD と零サプレス BDD の比較 (平均値)

	最大使用節点数	最大 BDD の節点数	最終 BDD の節点数	計算時間
従来の BDD	1.00	1.00	1.00	1.00
零サプレス BDD	0.48	0.41	0.187	0.85

表 10: 三つの乗算方法による BDD の節点数と計算時間の変化 (平均値)

乗算方法	従来の BDD			零サプレス BDD		
	最大使用節点数	最大 BDD の節点数	計算時間	最大使用節点数	最大 BDD の節点数	計算時間
a	1.00	1.00	1.00	0.42	0.36	0.73
b	0.70	0.69	0.75	0.33	0.28	0.62
c	0.49	0.42	0.42	0.30	0.23	0.38

Proc. of the 25th Design Automation Conference, pp.427-432, June 1988.

[10] 神田徳夫, 笹尾勤, "4 変数 AND-EXOR 最小論理式とその性質", 電子情報通信学会論文誌 D-I, Vol.J74-D-1, No.11, 1991, pp.765-773.

[11] 神田徳夫, 笹尾勤, "AND-EXOR 最小論理式の積項数の上界について", 電子情報通信学会論文誌 D-I, Vol.J75-D-1, No.3, 1992, pp.135-142.

[12] 神田徳夫, 笹尾勤, "下界定理を用いた AND-EXOR 論理式の簡単化法", 電子情報通信学会論文誌 D-I, Vol.J76-D-1,No.1, 1993, pp.1-10.

[13] 神田徳夫, 笹尾勤, "論理関数の LP 特徴ベクトルとその応用", 電子情報通信学会論文誌 D-I, Vol.J76-D-1,No.6, 1993, pp.260-268.

[14] N. Koda and T. Sasao, " LP equivalence class of logic functions", IFIP 10.5 Workshop on Application of the Reed-Muller expansion in Circuit Design, Sept. 1993.

[15] B. Lin and F. Somenzi, "Minimization of symbolic relations", *In Proc. of the IEEE International Conference on Computer Aided Design*, pp.88-91, Nov. 1990.

[16] S. Minato, N. Ishiura, and S. Yajima, "Shared binary decision diagram with attributed edges for efficient Boolean function manupulation", *Proc. 27th ACM/IEEE Design Automation Conf.*, June 1990, pp. 52-57.

[17] S. Minato, "Zero-suppressed BDDs for set manipulation in combinatorial problems", *Proc. 30th Design Automation Conference*, June 1993, pp. 272-277.

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

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

[20] 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.

[21] U. Rollwage, "The complexity of mod-2 sum PLA's for symmetric functions", IFIP 10.5 Workshop on Application of the Reed-Muller expansion in Circuit Design, Sept. 1993.

[22] K. K. Saluja and E. H. Ong, "Minimization of Reed-Muller canonic expansion", *IEEE Trans. Comput.*, C-28, pp. 535-537, 1979.

[23] 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.

[24] T. Sasao, "Logic synthesis with EXOR gates", in (Sasao e.d.) *Logic Synthesis and Optimization*, Kluwer Academic Publishers, 1993.

[25] T. Sasao, "AND-EXOR expressions and their optimization ", in (Sasao e.d.) *Logic Synthesis and Optimization*, Kluwer Academic Publishers, 1993.

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

[27] T. Sasao, "An exact minimization of AND-EXOR expressions using BDDs", IFIP 10.5 Workshop on Application of the Reed-Muller expansion in Circuit Design, Sept. 1993.