

Pseudoproduct に基づく回路とそのテスト容易性

石川 綾志[†] 五十嵐友則[†] 平山 貴司^{††} 清水 賢資[†]

[†] 群馬大学工学部情報工学科
〒 376-8515 群馬県桐生市天神町 1-5-1
^{††} 岩手大学情報システム工学科
〒 020-8551 盛岡市上田 4-3-5

E-mail: †{ishikawa,igarashi,shimizu}@ja4.cs.gunma-u.ac.jp, ††hirayama@cis.iwate-u.ac.jp

あらまし pseudocube の概念を用いた Sum-of-Pseudoproduct (SPP) 型は、EXOR 演算子を用いた新しい論理関数の表現法であり、従来の Sum-of-Products (SOP) 型よりもコンパクトな表現が可能である。本稿では、特に対称関数や実用回路において、SPP 型が SOP 型や ESOP 型よりも少ないリテラル数や積項数で実現可能であることを実験において示す。また、SPP 型が ESOP 型よりもテスト容易であること示し、よりテスト容易性を向上させるため、新たな表現法 EXOR-Sum-of-Pseudoproduct (ESPP) 型を提案する。この ESPP 型は、さらなるリテラル数の減少を導く可能性も持つ。

キーワード 排他的論理和, テスト容易化設計, ランダムパターンテスト, ESOP, pseudocube

Networks Based on Pseudoproduct and its Testability

Ryoji ISHIKAWA[†], Tomonori IGARASHI[†], Takashi HIRAYAMA^{††}, and Kensuke SHIMIZU[†]

[†] Department of Computer Science, Gunma University
Tenjincho 1-5-1, Kiryu-shi, Gunma, 376-8515 Japan
^{††} Department of Computer and Information Science, Iwate University
Ueda 4-3-5, Morioka-shi, Iwate, 020-8551 Japan

E-mail: †{ishikawa,igarashi,shimizu}@ja4.cs.gunma-u.ac.jp, ††hirayama@cis.iwate-u.ac.jp

Abstract Sum-of-Pseudoproduct (SPP) forms are new logic representations and have made possible to represent Boolean functions with more smaller expression than standard Sum-of-Products (SOP) forms. Our experimental results show that SPP forms require many fewer (pseudo)products and literals than SOP and ESOP for symmetric functions and practical networks. And, we propose EXOR-Sum-of-Pseudoproduct (ESPP) forms. These forms are EXOR-AND-EXOR ones and enhance the random pattern testability. Experimental results show that ESPP forms are more testable than SPP and ESOP. Furthermore they have new possibility of compactness of the networks.

Key words Exclusive-OR, design for testability, random pattern testing, ESOP, pseudocube

1. 序 論

Exclusive-OR (EXOR) を用いた論理設計の研究が行われている。実用回路のような特殊な性質を持つ回路において、EXOR 演算子を活用すると、従来の Sum-of-Product (SOP) 型などで実装するよりもゲート数を少なく見積もることができる。近年、この EXOR 演算子を用いた論理関数の表現法である pseudoproduct、そして Sum-of-Pseudoproduct (SPP) の研究が行われている [1] ~ [3]。

SPP 型は、実装の際に必要な積項数とリテラル数が、従来型よりも少ないことが知られている。本稿ではこれを示すために、すべての 3 変数関数と、4 変数 NPN 同値類の代表関数について、SOP、SPP、そして Exclusive-OR Sum-of-Product (ESOP) 型の積項数とリテラル数を調べた。ただし、SPP 型においては積項数は、pseudoproduct の数を表す。また、EXOR 演算子を用いた表現は、対称関数において SOP 型よりも少ない積項数を必要とすることが知られている [7]。実用回路の多くは、対称関数の性質を持っている。実験において、SPP 型の対称関数と MCNC ベンチマーク回路 [10] に対する効果を示す。

EXOR 演算子のもう一つの利点として、テスト容易性を高めることが挙げられる。現在、EXOR を用いた決定論的テストやランダムパターンテストに対する研究が行われている [14], [16]。本稿では、組込み型自己テストの普及や、テスト検査機器のクロックの向上などから盛んに用いられるようになってきた、ランダムパターンテストについて考察を行う。ESOP 型に対するテスト容易化設計は既に研究されている [12]。ランダムテストにおいて、ESOP 型は SOP 型よりもテスト容易ではないことが示されており、そのため因数分解によってランダムテスト容易性を向上させるなどの、付加的な要素が必要となってくる [8]。しかしながら、SPP 型についてはテスト容易性が未だ議論されていない。そこで本稿では、SPP 型のランダムテスト容易性について述べる。前述したように、SPP 型はよりコンパクトな表現が可能である場合が多い。そのため、故障の存在の有無を調べるポイントも少なくなるので、SPP 型のテストコストは低くなる。また、EXOR 演算子の使用と、その回路構造からテストコストを減少させる。実験結果では、SPP 型が従来型と比較してランダムテスト容易であることを示す。

さらに、pseudoproduct の概念に基づく新たな表現法を提案する。すなわち、EXOR-Sum-of-Pseudoproduct (ESPP) 型である。これは SPP 型

における出力段の OR 演算子を EXOR に置き換えたもので、pseudoproduct を EXOR で連結した型を持つ。ESPP 回路は EXOR ゲートを出力段に持つことによって、SPP 型よりもテスト容易である。これを実験によって示す。また、ESPP 型はよりコンパクトな回路になる可能性を持っている。

本稿の構成は以下の通りである。2 章では SPP 型の基本的な表記法と、最小化法について簡単に触れる。3 章でテスト容易性について述べ、4 章でよりテスト容易性を向上させる新たな表現法を提案する。5 章で実験結果を示し、最後に 6 章で結論と今後の課題について述べる。

2. 諸 定 義

この節では、pseudocube、pseudoproduct、Sum-of-Pseudoproduct (SPP) 型において必要な定義を示す。

SPP 型は subcube の概念を一般化した pseudocube に基づく。pseudocube は、次によって再帰的に定義される。ここで、 2^m 個の点からなる集合を degree m の pseudocube と呼ぶ。

(1) ブール n 次元空間 B^n における任意の一つの点は、degree 0 の pseudocube である。

(2) 任意の点の対は、degree 1 の pseudocube である。

(3) 2^m 個の点からなる集合 P が、二つの互いに素な degree $m-1$ の pseudocube P_1 と P_2 に分割でき、その変数の部分集合の補元をとることによって P_2 から P_1 を導出できる場合、 P は degree m の pseudocube である。

例として、図 1 は 2^3 個の点からなる degree 3 の pseudocube である。この行列は canonical 行列と呼ばれる。変数 x_0, x_2, x_4 は、この順序で列をソートした際にその行が 2 進数の昇順になるように選択されたもので、これを canonical 変数と呼び、それ以外を non-canonical 変数と呼ぶ。

pseudocube の特性関数は、pseudoproduct と呼ばれる。これは積項を一般化した概念であり、次で示される定義で生成される canonical 表現と呼ばれる EXOR-AND 型を持つ。

R を B^n 上の degree m の pseudocube とし、 M を P の canonical 行列とする。また、 $x_{p_0}, \dots, x_{p_{m-1}}$ と $x_{p_m}, \dots, x_{p_{n-1}}$ を、それぞれ canonical 変数と non-canonical 変数とする。このとき、pseudocube P の canonical 表現 $CEX(P)$ は、 $f_0 \cdot f_1 \cdot \dots \cdot f_{n-m-1}$ で与えられる。ここで、各 $f_i (0 \leq i \leq n-m-1)$ は、次の

	x_0	x_1	x_2	x_3	x_4	x_5
r_0	0	1	0	1	0	1
r_1	0	1	0	1	1	0
r_2	0	1	1	0	0	1
r_3	0	1	1	0	1	0
r_4	1	1	0	0	0	0
r_5	1	1	0	0	1	1
r_6	1	1	1	1	0	0
r_7	1	1	1	1	1	1

図1 2^3 個の点を持つ pseudocube の例

変数を含む EXOR factor である。

- (1) $M[0, p_{m+i}] \neq M[2^{m-j-1}, p_{m+i}]$ のとき、canonical 変数 $x_{p_j} (0 \leq j \leq m-1)$ 。
- (2) $M[0, p_{m+i}] = 1$ のとき、non-canonical 変数 $x_{p_{m+i}}$ 。 $M[0, p_{m+i}] = 0$ のとき、 $\bar{x}_{p_{m+i}}$ 。

□

図1において、列 x_i は変数 $x_i (0 \leq i \leq 5)$ に対応する。このとき、この pseudocube の canonical 表現は次のようになる。

$$CEX = x_1 \cdot (x_0 \oplus x_2 \oplus x_3) \cdot (x_0 \oplus x_4 \oplus x_5)$$

この例において、non-canonical 変数は x_1, x_3, x_5 の三つであるため、canonical 表現は三つの EXOR factor の積で与えられる。すなわち、 f_0 が $\hat{x}_1$ を含み、 f_1 が $\hat{x}_3$ を含み、 f_2 が $\hat{x}_5$ を含む。ここで、 $\hat{x}_i$ は x_i もしくは $\bar{x}_i$ を表す。任意の EXOR factor は canonical 変数を含む場合がある。ここでは、 f_1 が x_0 と x_2 を含み、 f_2 が x_0 と x_4 を含む。列 x_1 は、すべて1の定数ベクトルであるため、 f_0 は canonical 変数を含まない。この canonical 表現は、与えられた変数順序に対して一意に決定される。このようにして生成された表現は、従来法における積項に対応して、pseudoproduct と呼ばれる。

次に一般的な最小化アルゴリズムを示す[2]。従来の SOP 型の最小化は、主項の被覆問題に基づいている。この概念は直接 pseudoproduct に拡張でき、これを prime pseudoproduct と呼ぶ。prime pseudoproduct f は、 f 以外の任意の pseudoproduct によって被覆されることがない。pseudoproduct の集合が与えられたとき、次のアルゴリズムを使うことによって、最小化問題を解くことができる。

- (1) $\mathcal{P}$ を degree 0 (一つの点) の pseudocube の canonical 表現すべての集合とする。
- (2) 増加する $k (k = 0, 1, \dots)$ に対して、 $\mathcal{P}^+$ を

$\mathcal{P}$ の degree k の pseudoproduct の対から生成される degree $k+1$ の canonical 表現の集合とする。 $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}^+$ 。

$\mathcal{P}^-$ を h 個のリテラルを持つ degree k である canonical 表現の集合とする。ただし、それが高々 h リテラルを含む degree $m+1$ の canonical 表現を生成する場合、 $\mathcal{P} \leftarrow \mathcal{P} - \mathcal{P}^-$ 。

- (3) $\mathcal{P}$ からリテラル最小になるように被覆問題を解く。

□

以上のアルゴリズムを用いることで、最終的に論理式を pseudoproduct の論理和 (OR) で表現することが可能である。この表現を Sum-of-Pseudoproduct (SPP) 型と呼ぶ。

三段論理回路最小化問題を解くことは、SOP 型のような二段回路のそれよりも非常に困難である。しかしながら、三段論理は二段論理よりも低面積コストで実装が可能であることと、多段回路よりも高速なことから、研究されてきた。本稿で使用したプログラムでは、現在 10 変数程度の実用回路に対する canonical 表現の最小形、もしくは最小に近い形を生成可能である。

図2の例を従来の SOP 型で表現すると以下の様な式を持つ。 $x_1 x_2 x_4 + \bar{x}_0 x_1 \bar{x}_2 x_3 + x_0 x_1 \bar{x}_2 \bar{x}_3 + x_0 \bar{x}_1 \bar{x}_3 \bar{x}_4 + \bar{x}_0 \bar{x}_1 x_3 \bar{x}_4 + \bar{x}_0 x_1 x_2 \bar{x}_3 + x_0 x_1 x_2 x_3$ 。この場合、SOP 型で実現するために積項数 7、リテラル数 27 を必要とする。一方 SPP 型の場合、次のような式が生成され、pseudoproduct 数 2、リテラル数 8 で実装が可能である。 $x_1 \cdot (x_0 \oplus x_2 \oplus x_3) + (x_0 \oplus x_3) \cdot (x_1 \oplus \bar{x}_4)$ 。

		$x_0 x_1$			
		00	01	11	10
$x_2 x_3$	00			1	1
	01	1	1		
	11	1		1	
	10		1		1

$x_4=0$

		$x_0 x_1$			
		00	01	11	10
$x_2 x_3$	00			1	
	01		1		
	11		1	1	
	10		1	1	

$x_4=1$

図2 SOP 型と SPP 型における表現の例

3. SPP 型のランダムテスト容易性

この章では、各表現法に対するランダムパタンテストについて議論する。ランダムパターンは線形フィード

バックシフトレジスタやセルラオートマタなどのような単純なハードウェアで効果的に生成が可能であることから、組込み型自己テストにしばしば使用される。ランダムテストに対する EXOR 演算子を用いたテスト容易化設計が近年研究されている。これはパリティ関数である EXOR の使用が回路のテスト容易性を向上させる可能性を持つためである。

本稿における故障モデルは、一般的に広く用いられていることから単一縮退故障モデルとする。単一縮退故障は回路中の一つの信号線がある値 $v (v \in B, B = \{0, 1\})$ に固定されてしまう論理故障である。ここで、信号線 c の v 縮退故障を c/v で表すこととする。回路が故障であるか否かをテストする入力ベクトルをテストベクトルと呼ぶ。

最小テスト確率 P_{min} は、回路のテスト容易性の指標として幅広く用いられている [11]。ランダムテストパターンが回路に適用されるとき、故障 $c/0(c/1)$ を検出する確率を $c/0(c/1)$ の検出確率と呼び、 $P(c/0)(P(c/1))$ で表す。 H を検出可能な回路のすべての信号線の故障 $c/0, c/1$ の集合とする。このとき、 $\min_{h \in H} P(h)$ は回路の最小テスト確率と呼ばれ、 P_{min} で示される。

ESOP 型は EXOR 演算子を持つ表現の中では、さほどランダムテスト容易ではない。これを示すために、AND-EXOR 型の P_{min} を考える (図 3(a))。 n 変数関数を AND-EXOR 型で表現した場合、 n 入力 AND ゲートを持つことは少なくない。このような入力線 c の故障を検出することが一番難しい。すなわち、故障 $c/0$ や $c/1$ を検出するベクトルは、すべてのランダムパターン 2^n 個のうち、それぞれ一つずつしか存在しない。それゆえ、一般に AND-EXOR 型において、 $P_{min} \geq 1/2^n$ が成立する。実際に、すべての n 変数関数の半分は AND-EXOR 型で実装した場合に n 入力 AND ゲートを持つことが知られているので、 $P_{min} = 1/2^n$ が成立する。

EXOR 演算子は、AND-EXOR 型のように、従来回路の出力段に主に用いられてきた。一方、SPP 型は EXOR を回路の入力段に持ち、EXOR-AND-OR 型を生成する (図 3(b))。ここで、SPP 型のテスト容易性について、最小テスト確率を同様に用いて考える。

図 4 において、 A と B を SPP 型の AND ゲートとする。 A と B の入力をそれぞれ、 $c_1, c_2, \dots, c_j$ と $d_1, d_2, \dots, d_k$ で表す。値 j と k を、それぞれ A と B を含む pseudoproduct の non-canonical 変数の数と

する。 s を A の出力線とし、 P を s が接続する OR ゲートとする。同様に、 t を B の出力線とする。 R の出力線は e で示す。このとき、故障 $c_1/0$ を検出するテストは以下である。

$$c_1 \leftarrow 1, c_2, c_3, \dots, c_j \leftarrow 1, t \leftarrow 0$$

ランダムパターンが適用されたとき、 c_1 が 1 になる確率を $Pr[c_1 = 1]$ と表す。ここで c_1 が EXOR factor に接続されるので、 $Pr[c_1 = 1] = 1/2$ を得る。同様に、 $Pr[c_2 = 1] = Pr[c_3 = 1] = \dots = Pr[c_j = 1] = 1/2$ 。また、AND ゲート B の少なくとも一つの入力が 0 になるので、 $Pr[t = 0] = 1 - Pr[t = 1] = 1 - 1/2^k$ 。それゆえ、 $P(c_1/0)$ に対する次の式が成り立つ。

$$\begin{aligned} P(c_1/0) &= \frac{1}{2^j} \cdot (1 - \frac{1}{2^k}) \\ &= \frac{1}{2^j} - \frac{1}{2^{j+k}} \end{aligned}$$

$s/0$ と $e/0$ は、fault dominance [11] $P(e/0) \geq P(s/0) \geq P(c_1/0)$ から同等もしくはそれより確率的に高い値で故障の検出が可能である。同様に、 $P(c_1/1) = 1/2^j - 1/2^n$ かつ $P(e/1) \geq P(s/1) \geq P(c_1/1)$ が成立する。値 j は pseudoproduct の定義から、non-canonical 変数の数であり、それゆえ一般的に $j < n$ を得る。よって、SPP 型は AND-EXOR 型よりもテスト容易である。

SPP 型がテスト容易性を向上させるもう一つの理由として、回路をより簡単にする可能性を持つことが挙げられる。SPP 型は後に実験で示すが、他の表現法よりもブール関数をよりコンパクトに実現できる場合がある。このとき、テストすべき回路中のポイントも減少するため、テストのコストを下げる要因となる。

4. EXOR-Sum-of-Pseudoproduct

前章より、SPP 型は ESOP 型よりもランダムテスト容易であることが分かった。この章では、pseudoproduct に基づく SPP 型よりランダムテスト容易な新しい型を提案する。本稿において、それを EXOR-Sum-of-Pseudoproduct (ESPP) 型と呼ぶ。この型は、SPP 型の OR ゲートを EXOR ゲートに置き換えた表現であり、pseudoproduct の排他的論理和 (EXOR) の型になる (図 3(c))。入力段と出力段に EXOR を持つ ESPP 回路は、ランダムパターンテスト容易性をさらに改善する。

図 5 において、 A と B をそれぞれ ESPP 型の AND ゲートと EXOR ゲートとする。 A の入力線を $c_1, c_2, \dots, c_j$ で示す。値 j を A を含む

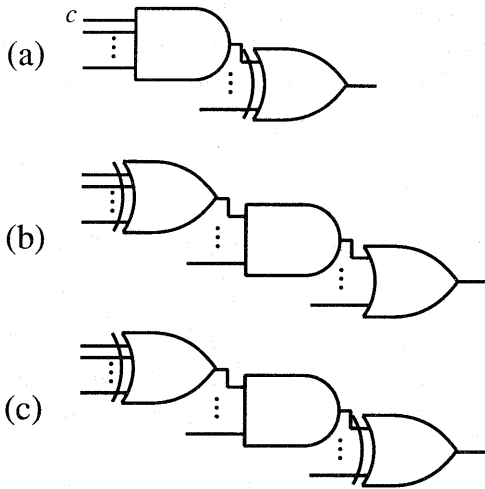


図3 (a) ESOP 型 (b) SPP 型 (c) ESPP 型

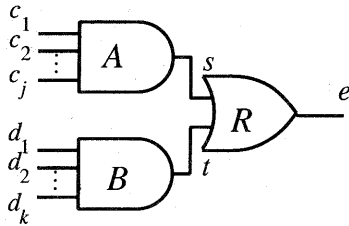


図4 AND-OR 回路

pseudoproduct の non-canonical 変数の数とする。s を A の出力線として、E を s が接続する EXOR ゲートとする。EXOR ゲート E の他の入力線を $t_1, \dots, t_k$ とする。また、E の出力線を e で示す。このとき、故障 $c_1/0$ を検出するテストは次である。

$$c_1 \leftarrow 1, c_2, c_3, \dots, c_j \leftarrow 1, \\ t_1, \dots, t_k \leftarrow \text{任意の入力組合せ}$$

EXOR ゲート E はパリティ関数であるため、故障 $c_1/0$ は $t_1, \dots, t_k$ の値を問わず出力へ伝播する。すなわち、この故障と $t_1, \dots, t_k$ の値は独立である。このとき、故障 $c_1/0$ に対する次の式が成り立つ。

$$P(c_1/0) = \frac{1}{2^j}$$

それゆえ、ESPP 型は ESOP 型や SPP 型よりもテスト容易である。

さらに、ESPP 型は新たな回路のコンパクト性を生み出す。例えば、図6を SPP 型で表すと、 $x_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 (x_0 \oplus \bar{x}_3) + \bar{x}_0 x_1 x_2$ となる。しかしながら、

ESPP 型で表した場合、 $(x_1 \oplus x_2)x_3 \oplus \bar{x}_0 x_2$ となり、リテラル数が減少することが分かる。

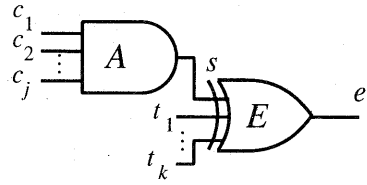


図5 AND-EXOR 回路

$x_0 x_1$	$x_2 x_3$			
	00	01	11	10
00				1
01		1	1	1
11		1		
10			1	

図6 ESPP 型の例

ESPP 型の最小化手続きは、SPP 型を求める手続きよりも非常に困難な問題である。ESPP 型に拘わらず、従来法の SOP 型や SPP 型においても、最小形を求める際には、大きく分けて二つの段階、すなわち主項を生成する段階と被覆問題を解く段階が存在する。SPP 型において、Ciriani は Extended Prime Pseudoproduct (EPPP) [3] の概念を示し、より正確に最小化を行うことを試みた。

EPPP は prime pseudoproduct の集合かつ、拡張から degree k の pseudocube が得られた際にリテラル数が減少しない場合は、その元となる degree $k-1$ の pseudoproduct を削除せずにそのまま残しておくようにした集合である。この EPPP を用いることによって、被覆時の pseudoproduct の候補を大幅に増大させ、より小さな表現を作る可能性を高められる。

しかしながら、この EPPP の欠点はその大幅に増える pseudoproduct の候補数である。被覆問題は計算コストが非常にかかるのは周知の通りであり、EPPP の概念は実用向きではない。そこで本稿では、2 章で与えた最小化アルゴリズムを改良して高速に SPP 型および ESPP 型の生成の足掛かりとする。この手続きは、degree 0 の pseudocube の集合が与えられたとき、隣接した pseudocube 同士を予め degree 1 の

pseudocube に拡張しておいて集合に加えておく。再帰的に拡張して degree を上昇させていく通常の pseudocube の生成手続きに対して、序盤にこの操作を行うことで計算コストを下げる役割をする。隣接した pseudocube の対は、実際に pseudoproduct を生成する際に、拡張されて一つの pseudoproduct に被覆される場合が多い。このため、大抵は拡張の精度を減少させることはない。また、被覆においては、pseudoproduct 数、リテラル数の順番で最小になるような SPP 型を選択する。これは一般に積項数の少ない方が実装のパフォーマンスに有利に働くからである。アルゴリズムは以下の通りである。

(1) P を degree 0 の pseudocube の canonical 表現すべての集合とする。

(2) P においてハミング距離が 1 の対から degree 1 の pseudocube の canonical 表現の集合 P_1 を生成する。このとき、生成された P_1 に被覆される degree 0 の pseudocube の canonical 表現を P_0 とする。

$$P \leftarrow P + P_1 - P_0$$

(3) 増加する k ($k = 0, 1, \dots$) に対して、 P^+ を P の degree k の pseudoproduct の対から生成される degree $k + 1$ の canonical 表現の集合とする。 $P \leftarrow P \cup P^+$ 。

P^- を h 個のリテラルを持つ degree k である canonical 表現の集合とする。ただし、それが高々 h リテラルを含む degree $m + 1$ の canonical 表現を生成する場合、 $P \leftarrow P - P^-$ 。

(4) P から pseudoproduct 数、リテラル最小の順に最小になるように被覆問題を解く。

□

pseudoproduct の EXOR を取るため、ESPP 型における被覆問題は SPP 型とは異なる。例えば図 6 の例では、実際に上記のアルゴリズムでは最小形を求めることができない。これは最小の ESPP 型が、 $(x_1 \oplus x_2)x_3$ と $\bar{x}_0x_2$ の EXOR となり、この各 pseudoproduct が生成不可能なためである。今回の実験ではテスト容易性を確かめることを主な目的とするので、最小形を求めるのではなく、ESPP 型の形になっており比較的小さな回路であればそれを採用した。ESPP 型の最小化アルゴリズムは、現在開発中である。

5. 実験結果

以上の議論から、従来の SOP、ESOP 型と SPP 型の表現法について実験を行った。計算に用いた CPU は、PentiumIII 550MHz(22.3 SPECint95) である。はじめに、3 変数関数のすべてと 4 変数の NPN 同値

類の代表関数について、各表現法の積項数とリテラル数を調べた (表 1)。表中の #P、#L、#PP はそれぞれ積項数、リテラル数、pseudoproduct 数を示している。SPP 型は、SOP の一般化した表現であるため、SOP よりも積項数、リテラル数が大きくなることはない。

表 1 SOP,ESOP,SPP 型における積項数とリテラル数の平均値

n	SOP		ESOP		SPP	
	#L	#P	#L	#P	#L	#PP
3	5.153	2.423	4.205	2.217	3.369	1.820
4	12.710	4.022	8.760	3.443	7.760	2.376

次に対称関数について各表現法の積項数とリテラル数を求めた (表 2)。対称関数では ESOP は SOP よりも少ない積項数、リテラル数で見積もることが可能であることが知られている [7]。実験から SPP 型は対称関数について、ESOP 型よりもさらに効果があることが分かった。同様にいくつかの MCNC ベンチマーク回路についても実験を行った (表 3)。ベンチマーク回路は多出力回路なので、各出力を一つの関数と見なした。表中の各回路名の括弧内の添字は、その回路の出力番号を示している。実験結果では、主に SPP 型がリテラル数、積項数ともに他の型よりも少なく見積もることができた。しかし、実用回路の性質によっては、評価が ESOP 型に有利であるもの (rd73(2) など) が存在した。

表 2 対称関数を実現するために必要な積項数の平均値

n	SOP	ESOP	SPP
3	2.625	2.250	1.500
4	5.156	4.031	2.560
5	10.125	6.969	3.694
6	20.055	12.093	5.508
7	39.563	21.418	13.583

また、各表現のテスト容易性を調べるために、MCNC ベンチマーク回路に対してランダムパターンテストを行った。表 4 は、検出可能なすべての 0 縮退故障と 1 縮退故障を検出するまでに必要なテストパタンの数 (テスト長) を示している。表中の TL はテスト長、time は計算時間を示している。テスト長はランダムパターンを 1000 回適用した結果の平均値である。

表3 MCNCベンチマーク回路における各表現の積項数とリテラル数

network	SOP		ESOP		SPP	
	#L	#P	#L	#P	#L	#PP
con1 (1)	11	4	29	9	11	4
con1 (2)	12	5	24	9	12	4
misex1 (1)	7	2	18	8	5	1
misex1 (2)	18	5	30	9	15	4
rd53 (1)	20	5	20	5	14	3
rd53 (2)	44	11	5	5	5	1
rd73 (1)	140	35	42	18	76	13
rd73 (2)	258	43	7	7	82	10
squar5 (1)	21	9	11	3	6	2
squar5 (2)	25	11	15	5	9	3
xor5 (1)	80	16	5	5	5	1

表4 MCNCベンチマーク回路における各表現のテスト長

回路名	ESOP		SPP		ESPP	
	TL	time	TL	time	TL	time
con1 (1)	69.3	10.0	45.5	7.82	32.0	5.24
con1 (2)	67.6	6.94	64.2	5.79	54.2	5.30
misex1 (1)	126.0	26.0	18.6	5.9	11.7	5.01
misex1 (2)	136.5	30.6	131.5	28.2	123.1	26.5
rd53 (1)	27.7	0.50	24.3	0.45	22.1	0.42
rd53 (2)	27.2	0.34	27.1	0.34	27.1	0.34
squar5 (1)	15.5	0.74	6.15	0.49	5.21	0.13
squar5 (2)	27.4	1.49	25.7	1.45	20.8	1.39

結果から、SPP型はESOP型よりテスト容易な場合が多いことが示された。特に、misex1(1)などは大幅にテスト容易性が向上した。また、提案したESPP型は、そのようなSPP型の多くをさらにテスト容易にすることが分かった。

6. 結 論

本稿では、従来のSOP型を一般化した概念であるSPP型について実験を行い、それがコンパクトな論理表現を可能とすることを示した。特に、対称関数やMCNCベンチマーク回路などの実用回路において、多大なる効果があることが分かった。また、SPP型が従来の型に比べテスト容易性であることに着目し、実験でそれを証明した。さらにテスト容易性を高めるため、pseudoproductに基づく新たな論理表現法であるESPP型を提案した。これはEXOR-AND-EXOR構造をしており、ランダムパターンテスト容易性を向上させる。実験結果より、SOPやSPP型よりもテスト容易であることが示された。

前述したように、ESPP型はさらなる回路のコンパクト化の可能性を持つ。それゆえ、今後の課題は最小なESPP型を導出するためのアルゴリズムを考えることである。

文 献

- [1] Fabrizio Luccio and Linda Pagli, "Normal Matrices, Pseudo-Cubes and Pseudo-Products," *Congressus Numerantium* Vol. 127 pp.33-56, 1997.
- [2] Fabrizio Luccio and Linda Pagli, "On a new Boolean function with applications," *IEEE Transactions on Computers*, Vol. 48, No. 3, pp. 296-310, 1999.
- [3] Valentina Ciriani, "Logic Minimization using Exclusive OR Gates," *Design Automation Conference 2001 Proc.*, pp.115-120, Jun. 2001.
- [4] Tsutomu Sasao, "Logic Synthesis and Optimization", Kluwer Academic Publishers, 1993.
- [5] Tsutomu Sasao, "Representations of Discrete Functions", Kluwer Academic Publishers, 1996.
- [6] Tsutomu Sasao, "Switching Theory for Logic Synthesis", Kluwer Academic Publishers, 1999.
- [7] Tsutomu Sasao and Philipp Besslich, "On the Complexity of Mod-2 Sum PLA's," *IEEE Transactions on Computers*, Vol. 39, No. 2, pp.262-266, Feb. 1990.
- [8] Ryoji Ishikawa, Takashi Hirayama, Goro Koda, and Kensuke Shimizu, "Factorization to Enhance Random Pattern Testability of EXOR Based Circuits," *IEEE APCCAS 2000 Proc.*, pp.795-798, Dec., 2000.
- [9] Tsutomu Sasao, "EXMIN2: A Simplification Algorithm for EXclusive-OR-Sum-of Products Expressions for Multiple-Valued-Input Two-Valued-Output Functions," *IEEE Trans. on Comput.*, Vol. 12, No. 5, pp. 621-632, May. 1993.
- [10] McElvain, K., "IWLS'93 benchmark set: version 4.0," Distributed as part of IWLS'93 benchmark distribution, May. 1993.
- [11] R. David, "Random Testing of Digital Circuits", Marcel Dekker, 1998.
- [12] Ugur Kalay, Douglas V. Hall, and Marek A. Perkowski, "A Minimal Universal Test Set for Self-Test of EXOR-Sum-of-Products Circuits," *IEEE Trans. Comput.*, vol. 49, no. 3, pp. 267-276, Mar. 2000.
- [13] H. Fujiwara, *Logic Testing and Design for Testability*, MIT Press, 1985.
- [14] R. Drechsler, H. Hengster, H. Schäfer, J. Hertmann, and B. Becker, "Testability of 2-level AND/EXOR circuits," *Proc. of the European Design and Test Conference '97*, pp. 548-553, Mar. 1997. bibitemChiGup94 C.-H. Chiang and S. K. Gupta, "Random pattern testable logic synthesis," *Proc. International Conference on Computer-Aided Design*, San-José, CA, pp. 125-128, 1994.
- [15] M. Chatterjee, D. K. Pradhan, and W. Kunz, "LOT: Logic optimization with testability— new transformations for logic synthesis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 5, pp. 386-399, May 1998.
- [16] T. Hirayama, G. Koda, Y. Nishitani, and K. Shimizu, "Easily testable realization based on single-rail-input OR-AND-EXOR expressions," *IEICE Trans. Inf. & Syst.*, vol. E82-D, no.9, pp. 1278-1286, Sep. 1999.