

# 論理関数のスペクトラル解析とその論理合成への応用に関する研究

九州工業大学

情報工学部・教授

笹尾 勤

〒 820-8502 飯塚市大字川津 680-4

2000 年 10 月 17 日

## 1 はじめに

論理関数のスペクトルを解析することにより、論理関数の大域的性質を抽出でき、それを用いて能率の良い回路を設計できる。スペクトラル解析をデジタルシステムの設計や解析に利用する手法は従来から知られていた [6]。しかし、 $n$  変数論理関数のスペクトルを計算するために  $2^n \times 2^n$  の行列を用いる計算法しか知られておらず、 $n = 30$  程度の問題でも、計算機の記憶容量を越えてしまい、実用上利用不可能であった。しかし、最近、MTBDD(多端子二分決定グラフ)を用いることにより、 $n = 100$  以上の論理関数のスペクトラムを短時間内に計算する方法が開発された。この方法を用いることにより、スペクトル解析を実用回路の設計に利用できる可能性が出てきた。本研究では、論理関数のスペクトルを解析することにより、論理関数の大域的性質を解析し、それを用いて回路を能率よく設計する方法について研究を行う。

与えられた々の関数を種々の論理式で表現した場合の表現の大きさを求める。次に、それらの大きさを簡単に推測する論理関数の尺度 (Measure) を論理関数のスペクトラムから求める。本研究では特に「AND-OR 回路で実現するのが得策か AND-EXOR 回路で実現するのが得策か」を高速に判定する尺度に関して、実験を行い統計的データを収集する。このデータを用いると、スペクトラムを調べることにより、いずれの方法で回路設計すべきかが明らかになる。

例えば、関数  $f = x_1 \oplus x_2 \oplus \dots \oplus x_n$  を表現するには、論理和形 (SOP: AND-OR 二段論理回路) では、 $2^{n-1}$  個の積項が必要であるが、EXOR 論理和形 (AND-EXOR 二段論理回路) では、わずか  $n$  個の積項で十分である。一方、関数  $f = x_1 y_1 \vee x_2 y_2 \vee \dots \vee x_n y_n$  を表現するには、SOP では  $n$  個の積項で十分であるが、EXOR 論理和形では、 $2^n - 1$  個の積項が必要となる。与えられた関数  $f$  を表現する EXOR 論理和形が SOP よりも遙かに簡単な場合には、EXOR に基づく論理設計法を使用すべきである。関数  $f$  を表現する最適な方法を見つけるには、一般に時間がかかる。従って、種々の表現法の大きさを高速に見積もる手段が必要である [2, 3, 7, 9, 17]。

本稿では、論理関数を実現する SOP 及び EXOR 論理和形の複雑度を高速に見積もるために尺度を提案する。そして、与えられた論理関数  $f$  を表現する固定極性リード・マラー 論理式 (FPRM) の積項数の平均値  $\eta(FPRM : f)$ 、とクロネッカー論理式 (KRO) の積項数の平均値  $\eta(KRO : f)$  を与える公式を導く。また、与えられた論理関数  $f$  を表現する、論理和形 (SOP) の複雑度の尺度  $\nu(f)$  についても述べる。 $\nu(f)$  は、最小論理和形の積項数の上界になっている。種々のベンチマーク関数に対

して、 $\eta(FPRM : f)$ 、 $\eta(KRO : f)$ 、および  $\nu(f)$  を示す。また、これらの値と  $\tau(FPRM : f)$ 、 $\tau(KRO : f)$ 、及び  $\tau(SOP : f)$  と比較する。ここで、 $\tau(FPRM : f)$  は、関数  $f$  を表現する最小 FPRM の積項数、 $\tau(KRO : f)$  は、最小 KRO の積項数、及び  $\tau(SOP : f)$  は、最小 SOP の積項数である。 $\nu(f) < \eta(KRO : f)$  のとき、多くの場合、 $f$  を表現する  $f$  を表現する SOP の積項数は、KRO の積項数よりも少なくなる。一方、 $\nu(f) > \eta(FPRM : f)$  または、 $\nu(f) > \eta(KRO : f)$  のとき、SOP の積項数は、FPRM や KRO の積項数よりも多くなる。ただし、例外も存在する。 $\eta(FPRM : f)$  や  $\eta(KRO : f)$  は、論理関数  $f$  の拡張真理値ベクトルや、EXOR\_TDD から計算でき、また、 $\nu(f)$  は、 $f$  の BDD から計算できる。後述するように、拡張真理値表 (EXOR\_TDD) は論理関数のスペクトラムと考えることができる。

## 2 積和形の複雑度の尺度

**定義 2.1** 論理関数  $f$  の最小論理和形の積項数を  $\tau(SOP : f)$  と表記する。他の論理式に対しても同様な記法を用いる。

SOP の場合、真の最小項の個数が、論理式の複雑度を見積もる一つの方法となっている。

**定義 2.2** 論理関数  $f$  の真の最小項の個数を  $\mu(f) = |f|$  と記す。

**補題 2.1**  $\tau(SOP : f) \leq \mu(f)$ 。

乱数関数に対しては、 $\mu(f)$  は複雑度の尺度として、かなり使える [2, 3, 7, 9, 17]。しかし、 $\mu(f) \geq 2^{n-1}$  の場合や、 $f(x_1, x_2, \dots, x_n) = x_1$  などの縮退関数に対して、 $\mu(f)$  は複雑度の尺度として、ほとんど無力である。

以下では、SOP の複雑度の尺度として、さらに精度の高いものを導く。

**定義 2.3** 論理関数  $f$  のシャノン展開  $f = x_i f_0 \vee x_i f_1$  を考える。ただし  $x_i$  は  $|f_0 f_1|$  の値が最大になるように選ぶ。このとき  $\nu(f) = |f| - |f_0 f_1|$  と定義する。ただし、 $f$  は  $n$  変数関数、 $f_0$  と  $f_1$  は  $(n-1)$  変数関数とする。

**定理 2.1**  $\tau(SOP : f) \leq \nu(f) \leq \mu(f)$ 。

(証明) シャノン展開  $f = \bar{x}_i f_0 \vee x_i f_1$  を考える。次に  $f = \bar{x}_i g_0 \vee x_i g_1 \vee g_2$  なる展開を考える。ここで  $g_0 = f_0 \bar{f}_1$ 、 $g_1 = \bar{f}_0 f_1$ 、 $g_2 = f_0 f_1$  である。このとき、 $\tau(SOP : f) \leq$

$\tau(SOP : g_0) + \tau(SOP : g_1) + \tau(SOP : g_2)$  である。また、 $\tau(SOP : g_0) \leq |g_0|$ 、 $\tau(SOP : g_1) \leq |g_1|$ 、 $\tau(SOP : g_2) \leq |g_2|$ 、 $|g_0| + |g_1| + 2|g_2| = \mu(f)$  が成立するので、次の結果を得る。

$$\begin{aligned} \tau(SOP : f) &\leq \nu(f) \\ &= \tau(SOP : g_0) + \tau(SOP : g_1) + \tau(SOP : g_2) \\ &= \mu(f) - |g_2|. \end{aligned}$$

(証明終)

定理 2.1 は、 $\nu(f)$  が、 $\mu(f)$  よりもタイトな上界であることを示している。論理関数  $f$  の真の最小項の個数は、 $f$  が BDD で表現されているときには、比較的容易に計算できる。

### 3 EXOR 論理和形の定義と性質

本節では、EXOR 論理和形の幾つかのクラスを定義し、その性質を紹介する [10, 12].

#### 3.1 正極性リード・マラー論理式 (PPRM)

次に示す展開は、EXOR 論理式の基礎となる [4, 12].

**補題 3.1** 任意の論理関数は、以下の三つの方法で展開可能である。

$$f = f_0 \oplus x_1 f_2, \quad (3.1)$$

$$f = f_1 \oplus \bar{x}_1 f_2, \quad (3.2)$$

$$f = x_1 f_0 \oplus x_1 f_1, \quad (3.3)$$

ここで

$$f_0 = f(0, x_2, x_3, \dots, x_n),$$

$$f_1 = f(1, x_2, x_3, \dots, x_n),$$

$$f_2 = f_0 \oplus f_1.$$

である。

(3.1)、(3.2)、および (3.3) を、それぞれ、正極性ダビオ展開、負極性ダビオ展開、シャノン展開と呼ぶ。(3.1) を関数  $f$  に再帰的に適用すると、次の補題が得られる。

**補題 3.2** 任意の  $n$  変数関数  $f(x_1, x_2, \dots, x_n)$  は、

$$\begin{aligned} f = & a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \dots \oplus a_n x_n \\ & \oplus a_{12} x_1 x_2 \oplus a_{13} x_1 x_3 \oplus \dots \oplus a_{n-1n} x_{n-1} x_n \oplus \\ & \dots \dots \dots \\ & \oplus a_{12\dots n} x_1 x_2 x_3 \dots x_n \end{aligned} \quad (3.4)$$

と表現できる。

(3.4) を正極性リード・マラー論理式 (PPRM) と呼ぶ。与えられた関数  $f$  に対して、上式の係数  $(a_0, a_1, a_2, \dots, a_{12\dots n})$  は、一意に決まる。従って、PPRM は標準形であり、PPRM の最小化問題は存在しない。論理式 (3.4) の積項数は高々  $2^n$  であり、すべてのリテラルは正極性 (肯定リテラル) である。

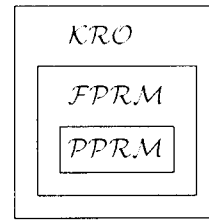


図 3.1: 種々の AND-EXOR 論理式の関係

#### 3.2 固定極性リード・マラー論理式 (FPRM)

各変数  $x_i$  ( $i = 1, 2, \dots, n$ ) に対して、正極性ダビオ展開、または負極性ダビオ展開のいずれかを適用して得られる論理式を固定極性リード・マラー論理式 (FPRM) という。各変数  $x_i$  に対して、正極性 ( $x_i$ ) と負極性 ( $\bar{x}_i$ ) の二つの極性が存在するので、 $n$  変数関数では、 $2^n$  個の異なる極性の集合が存在する。与えられた関数に対して、極性の集合が決まれば、係数の集合  $(a_0, a_1, \dots, a_{12\dots n})$  は一意に定まる。

#### 3.3 クロネッカー論理式 (KRO)

各変数に対して、シャノン展開、正極性ダビオ展開、または負極性ダビオ展開のいずれかを適用して得られる論理式をクロネッカー論理式 (KRO) という。各変数に対して、三つの展開法が存在するので、 $n$  変数関数に対して、高々  $3^n$  個の異なる KRO が存在する。

#### 3.4 種々の論理式の間関係

定義より、次の定理を得る。

**定理 3.1** PPRM, FPRM, および KRO を対応する論理式の集合とする。このとき、 $PPRM \subset FPRM \subset KRO$  が成立する。

図 3.1 に種々の AND-EXOR 論理式のクラスの間関係を示す。FPRM は、PPRM を真に含み、KRO の部分集合になっている。従って、任意の論理関数  $f$  に対して、次に定理を得る。

**定理 3.2**  $\tau(KRO : f) \leq \tau(FPRM : f) \leq \tau(PPRM : f)$ 。

#### 3.5 拡張真理値ベクトルとその表現

決定木は PPRM, FPRM, および KRO の生成に極めて重要である。本節では、EXOR 三分決定木 (ETDT) を紹介する。ETDT はこれら三つの論理式の簡単化に必要な情報をすべて包含する。ETDT の形式的な定義を述べる前に、簡単な例を示そう。

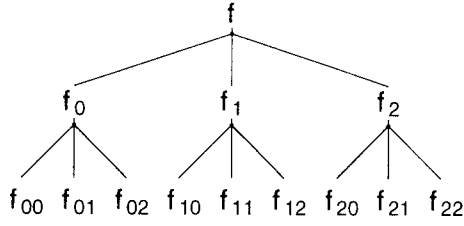


図 3.2: 2 変数関数の ETDT

例 3.1 2 変数関数  $f(x_1, x_2)$  を考える。図 3.2 に  $f(x_1, x_2)$  の完全 ETDT を示す。ここで、

$$\begin{aligned} f_0 &= f(0, x_2), & f_1 &= f(1, x_2), & f_2 &= f_0 \oplus f_1, \\ f_{00} &= f(0, 0), & f_{01} &= f(0, 1), & f_{02} &= f_{00} \oplus f_{01}, \\ f_{10} &= f(1, 0), & f_{11} &= f(1, 1), & f_{12} &= f_{10} \oplus f_{11}, \\ f_{20} &= f_{00} \oplus f_{10}, & f_{21} &= f_{01} \oplus f_{11}, & f_{22} &= f_{20} \oplus f_{21}. \end{aligned}$$

である。 (例終り)

以下に、 $n$  変数関数の完全 ETDT の形式的な定義を述べる。

定義 3.1 EXOR 三分決定木 (ETDT) では、頂点の集合  $V$  が非終端節点と終端節点の二つのタイプの節点から構成されている。非終端節点  $v$  は、属性として  $index(v) \in \{1, 2, \dots, n\}$  を持ち、三つの子供  $low(v), high(v), exor(v) \in V$  を持つ。終端節点  $v$  は、属性として  $value \in \{0, 1\}$  を持つ。ETDT と論理関数の間の関係は次のように定義される。

1.  $v$  が終端節点のとき:
  - (a)  $value(v) = 1$  ならば、 $f_v = 1$ .
  - (b)  $value(v) = 0$  ならば、 $f_v = 0$ .
2.  $v$  が非終端節点で  $index(v) = i$  のとき、 $f_v$  は、次に定義される論理関数である。

$$\begin{aligned} f_v(x_1, x_2, \dots, x_n) &= \\ &\bar{x}_i f_{low(v)}(x_1, x_2, \dots, x_n) \\ &\oplus x_i f_{high(v)}(x_1, x_2, \dots, x_n), \\ f_{EXOR(v)}(x_1, x_2, \dots, x_n) &= \\ &f_{low(v)}(x_1, x_2, \dots, x_n) \\ &\oplus f_{high(v)}(x_1, x_2, \dots, x_n). \end{aligned}$$

ETDT において、根から終端節点に至るどの経路においても、丁度  $n$  個の非終端節点を含むとき、ETDT は完全であるという。

与えられた関数が完全 ETDT で表現されているとき、ETDT の終端節点は拡張真理値ベクトルを表す。

定義 3.2  $n$  変数関数の拡張真理値ベクトルとは、 $3^n$  個の要素からなる二値ベクトルであり、各要素は完全 ETDT の終端節点に対応する。

例 3.2 図 3.2 に示す完全 ETDT を考える。4 個の要素からなる二値ベクトル  $[f_{00}, f_{01}, f_{10}, f_{11}]$  は、 $f(x_1, x_2)$  の真理値ベクトルである。また、9 個の要素からなる二値ベクトル  $[f_{00}, f_{01}, f_{02}, f_{10}, f_{11}, f_{12}, f_{20}, f_{21}, f_{22}]$  は、 $f(x_1, x_2)$  の拡張真理値ベクトルである。 (例終り)

次に、拡張真理値ベクトルの計算法について述べる。完全 ETDT の  $3^n$  個の終端節点は、拡張真理値ベクトル  $[f(\alpha_0), f(\alpha_1), \dots, f(\alpha_{3^n-1})]$  を表す。ここで、各要素は三値ベクトル  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ ;  $\alpha_i \in \{0, 1, 2\}$  ( $i = 1, 2, \dots, n$ ) で番号付けてある。

アルゴリズム 3.1 (拡張真理値ベクトル)

$i = 1, 2, \dots, n$  に対して、 $\alpha_i \in \{0, 1\}$  が成立するとき、 $f(\alpha)$  の値は、 $f$  の真理値表から得られる。また、ある  $i$  に対して、 $\alpha_i = 2$  のとき、 $f(\alpha)$  の値は、次のように計算できる。

$$\begin{aligned} f(\alpha_1, \dots, \overset{i}{2}, \dots, \alpha_n) &= f(\alpha_1, \dots, \overset{i}{0}, \dots, \alpha_n) \\ &\oplus f(\alpha_1, \dots, \overset{i}{1}, \dots, \alpha_n). \end{aligned}$$

拡張真理値ベクトルは、要素数  $3^n$  個のスペクトラムと考えることができる。これは、

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_{00} \\ f_{01} \\ f_{10} \\ f_{11} \end{bmatrix} = \begin{bmatrix} f_{00} \\ f_{01} \\ f_{02} \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{20} \\ f_{21} \\ f_{22} \end{bmatrix}$$

のような変換を考えたとき、最初の  $9 \times 4$  のマトリックスが、変換行列を定義し、拡張真理値ベクトル  $[f_{00}, f_{01}, f_{02}, f_{10}, f_{11}, f_{12}, f_{20}, f_{21}, f_{22}]$  がスペクトラムに対応する。この考え方は、一般に  $n$  変数関数に拡張できる [16]。

### 3.6 FPRM と KRO の極性ベクトル

定義 3.3  $n$  変数 FPRM の極性ベクトルとは、二値ベクトル  $\mathbf{a} = (a_1, a_2, \dots, a_n)$ ,  $a_i \in \{0, 1\}$  である。 $a_i = 0$  は、 $x_i$  で正極性ダビオ展開を使用することを示し、 $a_i = 1$  は、 $x_i$  で負極性ダビオ展開を使用することを示す。

$n$  変数関数の FPRM には、 $2^n$  個の異なる極性が存在し、極性ベクトルがその展開法を指定する。KRO の場合には、正極性ダビオ展開、負極性ダビオ展開、シャノン展開の三つの異なる展開法が存在するので、極性ベクトルは、 $3^n$  個の異なる極性が存在する。

定義 3.4  $n$  変数 KRO の極性ベクトルとは、三値ベクトル  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ ,  $\alpha_i \in \{0, 1, 2\}$  である。 $\alpha_i = 0$  は、 $x_i$  で正極性ダビオ展開を使用することを示し、 $\alpha_i = 1$  は、 $x_i$  で負極性ダビオ展開を使用することを示し、 $\alpha_i = 2$  は、 $x_i$  でシャノン展開を使用することを示す。

$n$  変数関数の KRO は、 $3^n$  個の異なる極性が存在し、極性ベクトル  $\alpha$  がその展開法を指定する。

$f$	$ext(f)$	
$f_{00}$	$f_{00}$	$f_{00}$
$f_{01}$	$f_{01}$	$f_{01}$
	$f_{00} \oplus f_{01}$	$f_{02}$
$f_{10}$	$f_{10}$	$f_{10}$
$f_{11}$	$f_{11}$	$f_{11}$
	$f_{10} \oplus f_{11}$	$f_{12}$
	$f_{00} \oplus f_{10}$	$f_{20}$
	$f_{01} \oplus f_{11}$	$f_{21}$
	$f_{00} \oplus f_{01} \oplus f_{10} \oplus f_{11}$	$f_{22}$

図 4.1: 拡張真理値表の導出

## 4 AND-EXOR 論理式の積項数の平均値

### 4.1 正極性リード・マラー論理式

定理 4.1 関数  $f$  を表現する PPRM の積項数は、

$$\tau(PPRM : f) = \sum_{\alpha \in \{0,2\}^n} f(\alpha).$$

で与えられる。ここで  $\sum$  は整数和を示す。

例 4.1  $n = 2$  の場合、

$$\tau(PPRM : f) = f(0,0) + f(0,2) + f(2,0) + f(2,2)$$

である。 (例終り)

定義 4.1 すべての  $n$  変数関数を表現する PPRM の積項数の平均値を  $\eta(PPRM : n)$  と記す。

定理 4.2  $\eta(PPRM : n) = 2^{n-1}$ 。

### 4.2 固定極性 リード・マラー論理式

定義 4.2 関数  $f$  を表現する FPRM の積項数の平均値を  $\eta(FPRM : f)$  と記す。

他の論理式のクラスに対しても、同様の記法を用いる。

定義 4.3  $f$  の拡張真理値ベクトルを  $ext(f)$  で記す。 $ext(f)$  の非零要素の個数を  $|ext(f)|$  で記す。つまり、 $|ext(f)| = \sum_{\gamma \in T^n} f(\gamma)$ 。ここで、 $T = \{0,1,2\}$  である。

例 4.2 図 4.1 に、 $n = 2$  変数関数の真理値ベクトルから拡張真理値ベクトルを導く方法を示す。真理値ベクトルは、 $2^n$  個の要素から構成される。一方、拡張真理値ベクトルは、 $3^n$  個の構成から要素される。 (例終り)

定理 4.3  $n$  変数関数  $f$  の FPRM の積項数の平均値は、

$$\eta(FPRM : f) = 2^{-n} \sum_{\gamma \in T^n} \lambda(\gamma) f(\gamma).$$

で与えられる。ここで  $\lambda(\gamma) = 2^k$ 、 $k$  は  $\gamma$  の 1 の個数であり、 $T = \{0,1,2\}$  である。

例 4.3  $n = 2$  のとき、

$$\begin{aligned} \eta(FPRM : f) = & 2^{-2} [(f(0,0) + f(0,1) + f(1,0) + f(1,1)) \\ & + 2(f(0,2) + f(1,2) + f(2,0) + f(2,1)) \\ & + 4f(2,2)]. \end{aligned}$$

(例終り)

定義より、次の関係を得る。

系 4.1  $\tau(FPRM : f) \leq \eta(FPRM : f)$ 。

### 4.3 クロネッカー論理式 (KRO)

定理 4.4  $n$  変数関数  $f$  の KRO の積項数の平均値は、

$$\eta(KRO : f) = (2/3)^n \sum_{\gamma \in T^n} f(\gamma).$$

で与えられる。ここで、 $T = \{0,1,2\}$  である。

例 4.4  $n = 2$  の場合、次のようになる。

$$\begin{aligned} \tau(f) = & (2/3)^2 [f(0,0) + f(0,1) + f(0,2) \\ & + f(1,0) + f(1,1) + f(1,2) \\ & + f(2,0) + f(2,1) + f(2,2)]. \end{aligned}$$

(例終り)

定義より、次の関係を得る。

系 4.2  $\tau(KRO : f) \leq \eta(KRO : f)$ 。

## 5 複雑度の尺度の計算法

### 5.1 データ構造とアルゴリズム

本節では、EXOR\_TDD (EXOR 三分決定グラフ) を用いて拡張真理値ベクトルを表現する方法について述べる。 $n$  の値が大きいとき、本手法で必要なメモリ量や計算時間は、拡張真理値ベクトルを率直に表現するよりも少なくできる。

定義 5.1 入力変数の順序が固定されていると仮定する。関数  $f$  の EXOR 三分決定グラフ (EXOR\_TDD) とは、 $f$  の EXOR 完全三分決定木において、同型な部分グラフを共有して得られる有向グラフである。

EXOR\_TDD[10] は、二分決定グラフ (BDD)[1] と同様なアルゴリズムで生成できる。入力変数の個数が多い時には、BDD は、真理値ベクトルや、SOP のキューブ表現よりもコンパクトに (少ないメモリで) 表現できる。同様に、EXOR\_TDD も、拡張真理値ベクトルよりコンパクトに表現できる。\$n\$ 変数関数の完全三分決定木の節点数は、\$1 + 3^1 + 3^2 + \dots + 3^{n-1} = (3^n - 1)/2\$ で与えられる。しかし、EXOR\_TDD では、一つの部分関数に対して、ただ一つの部分グラフを実現すればよいので、同じ部分関数が何度も生じる場合には、節点数を削減できる。

**定理 5.1** 任意の \$n\$ 変数関数は、節点数 \$O(3^n/n)\$ の EXOR\_TDD で表現可能である。

**定理 5.2** 任意の \$n\$ 変数対称関数は、節点数 \$O(n^3)\$ の EXOR\_TDD で表現可能である。

**アルゴリズム 5.1** \$\eta(FPRM : f)\$ の計算

1. 関数 \$f\$ の EXOR\_TDD を構成する。
2. 各枝に重み \$\lambda(\gamma)\$ を付ける。各パスの重みの和を計算する。
3. \$\eta(FPRM : f) = 2^{-n} \sum\_{\gamma \in T^n} \lambda(\gamma) f(\gamma)\$。

**アルゴリズム 5.2** \$\eta(KRO : f)\$ の計算

1. 関数 \$f\$ の EXOR\_TDD を構成する。
2. 1 枝に重み 1, 0 枝に重み 0 を付け、1-path の重みの計算を行う。
3. \$\eta(KRO : f) = (2/3)^n \sum\_{\gamma \in T^n} f(\gamma)\$。

## 5.2 多出力関数への拡張

与えられた \$n\$ 入力 \$m\$ 出力関数 \$\mathbf{f} = (f\_0, f\_1, \dots, f\_{m-1})\$ が多端子二分決定グラフ (MTBDD)[12] で表現されているものとする。この場合、複雑度の尺度を次のように計算する。\$\mathbf{f}\$ に対する MTTDD と、\$ext(\mathbf{f})\$ に対する MTBDD を構成する。

$$\begin{aligned} \mu(\mathbf{f}) &= \text{非零要素に至る経路の数} \\ &= 2^n - (\text{零要素に至る経路の数}) \\ \nu(\mathbf{f}) &= \mu(\mathbf{f}) - (\mathbf{f}(0, \mathbf{a}) = \mathbf{f}(1, \mathbf{a}) \neq \mathbf{o}) \\ &\quad \text{となるような要素 } \mathbf{a} \in B^{n-1} \text{ の個数.} \\ \eta(FPRM : \mathbf{f}) &= 2^{-n} \sum_{\gamma \in T^n} \lambda(\gamma) (f(\gamma) \neq \mathbf{o}) \\ \eta(KRO : \mathbf{f}) &= \left(\frac{2}{3}\right)^n \sum_{\gamma \in T^n} (f(\gamma) \neq \mathbf{o}) \end{aligned}$$

## 6 実験結果

表 6.1 に種々の関数に対する実験結果を示す。これらの関数は 4 つのグループに分類できる。

### 6.1 AND-EXOR 論理式で表現された関数

\$SB(n, 1) = x\_1 \oplus x\_2 \oplus \dots \oplus x\_n\$ は、\$n\$ 変数パリティ関数である。この関数を表現するには FPRM と KRO では、\$n\$ 積項で十分であるが、SOP では \$2^{n-1}\$ 積項必要である。

\$SB(9, 4) = x\_1 x\_2 x\_3 x\_4 \oplus x\_1 x\_2 x\_3 x\_5 \oplus \dots \oplus x\_6 x\_7 x\_8 x\_9\$ は、ESOP 簡単化プログラムのベンチマーク関数である [8]。この関数 \$f\$ では、関係 \$\tau(SOP : f) = \tau(FPRM : f) = \tau(KRO : f)\$ が成立している。

Test1 は関数 \$f = x\_1 x\_2 \dots x\_n \oplus x\_{n+1}\$, (\$n = 8\$) である。すべての \$n\$ に対して、\$\tau(FPRM : f) = \tau(KRO : f) = 2\$、\$\tau(SOP : f) = n + 1\$ が成立する。

### 6.2 AND-OR 論理式で表現された関数

Test2 は関数 \$f = \bar{x}\_1 \bar{x}\_2 \dots \bar{x}\_n \vee x\_1 x\_2 \dots x\_n\$, (\$n = 2r = 8\$) である。すべての \$n\$ に対して、\$\tau(SOP : f) = 2\$、\$\tau(FPRM : f) = 2^{r+1} - 2\$、\$\tau(KRO : f) = 2\$ が成立する。

Test3 は関数 \$f = x\_1 x\_2 \vee x\_3 x\_4 \vee \dots \vee x\_{n-1} x\_n\$, (\$n = 2r = 8\$) である。すべての \$n\$ に対して、\$\tau(SOP : f) = r\$、\$\tau(FPRM : f) = \tau(KRO : f) = 2^r - 1\$ が成立する。

### 6.3 算術演算関数

adr4, inc8, log8, mlp4, nrm4, rdm8, rot8, sqr8, wgt8, and sym9 など算術演算関数である。このうち、adr4, inc8, mlp4, rdm8, sqr8, および wgt8 では、KRO の積項数は SOP の積項数よりも少ない。これらの場合、\$\eta(KRO : f) < \nu(f)\$ となっている。

### 6.4 乱数関数

\$n = 7, 8, 9\$ に対して、疑似乱数関数を生成した。ただし、どの関数も真の最小項を丁度 \$2^{n-1}\$ 個含む。どの場合にも、SOP の積項数は、KRO や FPRM よりも少なくなった。これらの場合、\$\nu(f) < \eta(KRO : f)\$、\$\nu(f) < \eta(FPRM : f)\$ という関係になっている。

## 7 結論

本稿では、論理関数 \$f\$ を表現する FPRM および KRO の積項数の平均値を与える公式 \$\eta(FPRM : f)\$ および \$\eta(KRO : f)\$ を導いた。これらの値は、\$f\$ の拡張真理値ベクトルや EXOR\_TDD から容易に計算できる。次に、種々のベンチマーク関数に対して、\$\eta(FPRM : f)\$ と \$\eta(KRO : f)\$ の値を計算し、\$f\$ の最小 FPRM の積項数 \$\tau(FPRM : f)\$ および最小 KRO の積項数 \$\tau(KRO : f)\$ と比較した。また、SOP の複雑度の尺度として \$\nu(f)\$ を定義した。これは、\$f\$ の最小 SOP の積項数の上界になっている。\$f\$ が BDD で表現されている場合、\$\nu(f)\$ も容易に計算できる。種々のベンチマーク関数に対して、\$\nu(f)\$ の値も求めた。多くの場合、\$\nu(f) < \eta(KRO : f)\$ のとき、SOP の積項数は、KRO の積項数よりも少ない。一方、\$\nu(f) > \eta(FPRM : f)\$ または \$\nu(f) > \eta(KRO : f)\$ の

表 6.1: 実験結果

	入力	出力	$\eta$ : 平均値			$\nu$	$\tau$ : 最小値		
			FP RM	KRO	SOP		FP RM	KRO	SOP
SB(8, 1)	8	1	8.5	44.9	128	8	8	128	
SB(9, 1)	9	1	9.5	66.6	256	9	9	256	
SB(9, 4)	9	4	189.0	259.4	218	126	126	126	
Test1	9	1	27.1	26.6	129	2	2	9	
Test2	8	1	49.3	19.9	2	30	2	2	
Test3	8	1	57.6	54.2	101	15	15	4	
adr4	8	5	70.0	121.7	255	34	34	75	
inc8	8	9	56.2	79.9	256	16	16	37	
log8	8	8	230.7	222.9	186	193	171	123	
mlp4	8	8	205.8	204.2	225	97	97	121	
nm4	8	5	226.6	211.2	210	185	157	120	
rdu8	8	8	78.0	121.8	255	56	56	76	
rot8	8	5	175.6	154.8	143	118	83	57	
sqr8	8	16	236.8	239.2	255	168	168	178	
wgt8	8	4	154.0	202.1	255	107	107	255	
sym9	9	1	186.5	251.2	238	173	173	84	
Random7	7	1	65.7	64.4	44	54	48	23	
Random8	8	1	126.5	127.0	93	101	100	47	
Random9	9	1	250.4	254.7	185	226	212	88	

とき、SOPの積項数は、FPRMやKROの積項数よりも多い。もちろん、Test3のように例外もある。 $\eta(FPRM : f)$ 、 $\eta(KRO : f)$ 、 $\nu(f)$ は比較的容易に計算できるので、 $\tau(FPRM : f)$ 、 $\tau(KRO : f)$ 、 $\tau(SOP : f)$ を見積もるのに利用できる。

本論文の基礎となっている拡張真値ベクトルやEXOR\_TDDは、論理関数のスペクトラムの一種と考えられ[16]、スペクトラムから論理式の複雑度がある程度推測できることが分かる。

## 謝辞

本研究は、一部、高柳記念電子科学技術振興財団の研究助成および、文部省科学研究費補助金による。

## 参考文献

- [1] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. Comput.*, Vol. C-35, No. 8, Aug. 1986, pp. 677-691.
- [2] K. Cheng and V. D. Agrawal, "An entropy measure for the complexity of multi-output Boolean functions," *Design Automation Conference*, pp. 302-305, June 1990.
- [3] R. W. Cook and M. J. Flynn, "Logical network cost and entropy," *IEEE Trans. Comput.*, Vol. C-22, No. 9, pp. 823-826, Sept. 1973.

- [4] M. Davio, J-P. Deschamps and A. Thayse, *Discrete and switching functions*, McGraw-Hill International, 1978.
- [5] D. Debnath and T. Sasao, "Fast Boolean matching under variable permutation using representative," *Asia and South Pacific Design Automation Conference, ASP-DAC'99*, pp. 359-362, Jan. 1999.
- [6] S. L. Hurst, D. M. Miller, and J. C. Muzio, *Spectral Techniques in Digital Logic*, Academic Press, 1985.
- [7] F. Mileto and G. Putzolu, "Average values of quantities appearing in Boolean function minimization," *IEEE Trans. Elec. Comput.*, Vol. EC-13, No. 4, April 1964, pp. 87-92.
- [8] T. Sasao and P. Besslich, "On the complexity of MOD-2 Sum PLA's," *IEEE Trans. on Comput.*, Vol. 32, No. 2, pp. 262-266, Feb. 1990.
- [9] T. Sasao, "Bounds on the average number of products in the minimum sum-of-products expressions for multiple-valued input two-valued output functions," *IEEE Trans. on Comput.*, Vol. 40, No. 5, pp. 645-651, May 1991.
- [10] T. Sasao (ed.), *Logic Synthesis and Optimization*, Kluwer Academic Publishers, 1993.
- [11] 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.
- [12] T. Sasao and M. Fujita (ed.), *Representation of Discrete Functions*, Kluwer Academic Publishers, 1996.
- [13] T. Sasao and D. Debnath, "Generalized Reed-Muller expressions: Complexity and an exact minimization algorithm," *IEICE Transactions Fundamentals*, Vol. E79-A, No. 12, pp. 2123-2130, Dec. 1996.
- [14] T. Sasao, "Arithmetic ternary decision diagrams and their applications," *Fourth International Workshop on Applications of the Reed-Muller Expansion in Circuit Design, (Reed-Muller 99)*, Victoria, Canada, August 20-21, 1999.
- [15] T. Sasao and K. Kurimoto, "Three parameters to find functional decompositions," *Asia and South Pacific Design Automation Conference (ASP-DAC'2000)*, pp. 259-264, Yokohama, Japan, Jan. 26-28, 2000.
- [16] R. S. Stanković, and T. Sasao, "Spectral Interpretation of TDDs," *the Seventh Workshop on Synthesis And System Integration of Mixed Technologies (SASIMI'97)*, Osaka, Japan, Nov. 25-26, 1997.
- [17] D. Varma and E. A. Trachtenberg, "On the estimation of logic complexity for design automation applications," in *1990 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pp. 368-71, Oct. 1990.
- [18] S. Yang, "Logic synthesis and optimization benchmark user guide, version 3. 0," *MCNC*, Jan. 1991.

## Publication list 1998–2000

### 1988

- [1] R. Stankovic and T. Sasao, "Decision diagrams for discrete functions: Classification and unified interpretation," *Asia and South Pacific Design Automation Conference, ASP-DAC'98*, Feb. 1998.
- [2] D. Debnath and T. Sasao, "A heuristic algorithm to design AND-OR-EXOR three-level networks," *Asia and South Pacific Design Automation Conference, ASP-DAC'98*, Feb. 1998.
- [3] Hafiz Md. Hasan Babu and T. Sasao, "Design of multiple-output networks using time domain multiplexing and shared multi-terminal multiple valued decision diagrams," *IEEE International Symposium on Multiple-Valued Logic*, pp. 45–51, Fukuoka, Japan, May 27–29, 1998.
- [4] J. Butler and T. Sasao, "On the properties of multiple-valued functions that are symmetric in both variable values and labels," *IEEE International Symposium on Multiple-Valued Logic*, pp. 83–88, Fukuoka, Japan, May 27–29, 1998.
- [5] T. Sasao and M. Matsuura, "DECOMPOS: An integrated system for functional decomposition," *ACM/IEEE International Workshop on Logic Synthesis*, Lake Tahoe, CA, June 1998.
- [6] Y. Iguchi, T. Sasao, and M. Matsuura, "On properties of Kleene TDDs," *IEICE Trans. Information and Systems*, Vol. E81-D, No. 7, pp. 716–723, July. 1998.
- [7] Hafiz Md. Hasan Babu and T. Sasao, "Representations of multiple-output logic functions by binary decision diagrams for characteristic functions," *the Eighth Workshop on Synthesis And System Integration of Mixed Technologies (SASIMI'98)*, pp. 101–108, Sendai, Japan, Oct. 19–20, 1998.
- [8] Hafiz Md. Hasan Babu and T. Sasao, "Shared multi-terminal binary decision diagrams for multiple-output functions," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E81-A, No. 12, pp. 2545–2553, Dec. 1998.
- [9] Hafiz Md. Hasan Babu and T. Sasao, "Time-division multiplexing realizations of multiple-output functions based on shared multi-terminal multiple-valued decision diagrams," *IEICE Trans. Information and Systems*, Vol. E82-D, No. 5, pp. 925–932, May 1999.
- [10] T. Sasao and S. Kajihara, "Functional decompositions using an automatic test pattern generator and a logic simulator," *ACM/IEEE International Workshop on Logic Synthesis*, Lake Tahoe, CA, June 1999.
- [11] T. Sasao, "Arithmetic ternary decision diagrams and their applications," *Fourth International Workshop on Applications of the Reed-Muller Expansion in Circuit Design, (Reed-Muller 99)*, Victoria, Canada, August 20–21, 1999.
- [12] D. Debnath and T. Sasao, "Exact minimization of FPRMs for incompletely specified logic functions," *Fourth International Workshop on Applications of the Reed-Muller Expansion in Circuit Design, (Reed-Muller 99)*, Victoria, Canada, August 20–21, 1999.
- [13] Hafiz Md. Hasan Babu and T. Sasao, "Representations of multiple-output functions using binary decision diagrams for characteristic functions," *IEICE Trans. Fundamentals*, Vol. E82-A, No. 11, pp. 2398–2406, Nov. 1999.

### 1999

- [1] Y. Iguchi, T. Sasao, M. Matsuura, and A. Iseno, "Realization of regular ternary logic functions using double-rail logic," *Asia and South Pacific Design Automation Conference, ASP-DAC'99*, pp. 331–334, Jan. 1999.
- [2] D. Debnath and T. Sasao, "Fast Boolean matching under variable permutation using representative," *Asia and South Pacific Design Automation Conference, ASP-DAC'99*, pp. 359–362, Jan. 1999.
- [3] T. Sasao, *Switching Theory for Logic Synthesis*, Kluwer Academic Publishers, 1999.
- [4] T. Sasao, "Totally undecomposable functions: applications to efficient multiple-valued decompositions," *IEEE International Symposium on Multiple-Valued Logic*, pp. 59–65, Freiburg, Germany, May 20–23, 1999.
- [5] Hafiz Md. Hasan Babu and Tsutomu Sasao, "Shared multiple-valued decision diagrams for multiple-output functions," *IEEE International Symposium on Multiple-Valued Logic*, pp. 166–172, Freiburg, Germany, May 20–23, 1999.
- [6] D. Debnath and T. Sasao, "Multiple-valued minimization to optimize PLA with output parity gates," *IEEE International Symposium on Multiple-Valued Logic*, pp. 99–104, Freiburg, Germany, May 20–23, 1999.
- [7] T. Sasao and K. Kurimoto, "Three parameters to find functional decompositions," *Asia and South Pacific Design Automation Conference (ASP-DAC'2000)*, pp. 259–264, Yokohama, Japan, Jan. 26–28, 2000.
- [8] D. Debnath and T. Sasao, "Exact minimization of fixed polarity Reed-Muller expressions for incompletely specified functions," *Asia and South Pacific Design Automation Conference (ASP-DAC'2000)*, pp. 247–252, Yokohama, Japan, Jan. 26–28 2000.
- [9] Y. Iguchi, T. Sasao, M. Matsuura, and A. Iseno, "A hardware simulation engine based on decision diagrams," *Asia and South Pacific Design Automation Conference (ASP-DAC'2000)*, pp. 73–76, Yokohama, Japan, Jan. 26–28, 2000.
- [10] Hafiz Md. Hassan Babu and Tsutomu Sasao, "Minimization of multiple-valued decision diagrams using sampling method," *Proceedings of the Synthesis and System Integration of Mixed Technologies (SASIMI 2000)*, Kyoto, Japan, April 6–7, 2000.
- [11] Hafiz Md. Hasan Babu and T. Sasao, "Representations of multiple-output switching functions using multiple-valued pseudo-Kronecker decision diagrams," *30th International Symposium on Multiple-Valued Logic*, Portland, Oregon, U.S.A., May 23–25, 2000.
- [12] T. Sasao, "On the number of dependent variables for incompletely specified multiple-valued functions," *30th International Symposium on Multiple-Valued Logic*, Portland, Oregon, U.S.A., May 23–25, 2000.
- [13] Y. Iguchi, T. Sasao, and M. Matsuura, "Implementation of multiple-output functions using PROMDDs," *30th International Symposium on Multiple-Valued Logic*, Portland, Oregon, U.S.A., May 23–25, 2000.
- [14] T. Sasao, "A New expansion of symmetric functions and their application to non-disjoint functional decompositions for LUT-type FPGAs," *International Workshop on Logic Synthesis*, Dana Point, California, U.S.A., May 31 – June 2, 2000.