

EVMDDD に基づく二変数数学関数の表現法と数値計算回路への応用

EVMDDD-Based Representations of Two-Variable Mathematical Functions and Their Applications to Function Generators

永山 忍¹ 笹尾 勤²Shinobu NAGAYAMA¹ Tsutomu SASAO²¹広島市立大学 情報科学部 情報工学科²九州工業大学 情報工学部 電子情報工学科¹Dept. of Computer and Network Engineering, Hiroshima City University²Dept. of Computer Science and Electronics, Kyushu Institute of Technology

概要 本稿は、多値決定グラフを用いた二変数数学関数の表現法と数値計算回路の設計法を提案する。二変数数学関数を表現する EVMDDD (Edge-Valued Multi-valued Decision Diagram) がコンパクトであることを示すために、本稿では、まず、 l 限定 M_p 単調増大関数という関数のクラスを定義し、 l 限定 M_p 単調増大関数を表現する EVBDD (Edge-Valued Binary Decision Diagram) の節点数の上界を導出する。そして、 l 限定 M_p 単調増大関数を表現する EVBDD は、 p が小さいとき、他の表現法に比べコンパクトであることを示す。実験に用いた二変数数学関数は、 $p=1$ または $p=3$ の l 限定 M_p 単調増大関数に変換でき、EVBDD でコンパクトに表現できることを示す。また、EVMDDD を用いることで、決定グラフのメモリ量やパス長をさらに削減できることを示す。EVMDDD の値を評価する順序回路のアーキテクチャを提案し、いくつかの二変数数学関数の FPGA 実装結果により、提案手法の有用性を示す。

1 はじめに

コンピュータグラフィックスやデジタル信号処理などの様々な分野で、数学(初等)関数は、加算や乗算と同様に基本的な演算として広く利用されており、数学関数を計算する様々な回路の設計法が提案されている[18]。しかし、ほとんどの既存手法[5, 16, 22, 27, 28]が一変数数学関数のみを対象としており、多変数(二変数以上の)数学関数を対象とした回路の設計法は、あまり報告されていない[7, 8, 31]。一般的なデジタル回路の設計には、BDD (Binary Decision Diagram) などの決定グラフがしばしば利用されており[6, 17]、論理関数や整数関数のグラフ表現法に関する様々な研究[24, 33, 34]が報告されている。しかし、数学関数のグラフ表現法に関する研究はあまり報告されておらず[21, 26, 30]、特に、多変数数学関数のグラフ表現法に関する研究は、私たちの知る限り報告されていない。

そこで本稿では、EVMDDD (Edge-Valued Multi-valued Decision Diagram) を用いた二変数数学関数の表現法と EVMDDD に基づく数値計算回路の構成および設計法を提案

する。二変数数学関数の複雑さを解析するために、本稿では、 l 限定 M_p 単調増大関数という関数のクラスを定義し、 l 限定 M_p 単調増大関数を表現する EVBDD (Edge-Valued Binary Decision Diagram) の節点数の上界を示す。理論的解析と実験結果により、EVMDDD は、一変数数学関数だけでなく二変数数学関数もコンパクトに表現でき、EVMDDD を用いることで一変数数学関数と二変数数学関数を同じ回路構成で実現できることを示す。

本稿は、以下のように構成されている。第 2 節では、実数(数学)関数を整数関数へ変換するための固定小数点表現と本稿で使用する決定グラフを定義する。第 3 節では、決定グラフを用いた二変数数学関数の表現法について述べる。ここでは、 l 限定 M_p 単調増大関数を定義し、 l 限定 M_p 単調増大関数を表現する EVBDD の節点数の上界を示す。そして第 4 節で、EVMDDD に基づく数値計算回路の構成と設計法について述べる。

2 諸定義

2.1 数値表現と精度

本節では、実数関数を整数関数へ変換するための数値表現と精度を定義する。

定義 1. $B = \{0, 1\}$, Z を整数の集合, そして R を実数の集合とする. n 入力 m 出力論理関数 (多出力論理関数) は写像 $B^n \rightarrow B^m$, 整数関数は写像 $B^n \rightarrow Z$, そして, 二変数数学 (実数) 関数は写像 $R \times R \rightarrow R$ である.

定義 2. 二進固定小数点で表現された数値 X を

$$X = (x_{s-1} x_{s-2} \dots x_0 . x_{-1} x_{-2} \dots x_{-t})_2$$

と表記する. ただし, $x_i \in \{0, 1\}$, s は整数部のビット数, t は小数部のビット数である. このとき X は, 以下の式で計算できる.

$$X = \sum_{i=-t}^{s-1} 2^i x_i$$

本稿では, X に含まれる二値変数の集合を $\{X\}$ で表す.

定義 3. 精度は, 実数計算における有効桁数を意味する. 特に, n ビット精度とは, 固定小数点表現された実数計算において有効桁数が n ビット, すなわち, $n = s + t$ を意味する. 本稿で, n ビット精度関数 $f(X, Y)$ とは, 変数 X と Y がそれぞれ n ビット精度の関数であることを意味する.

固定小数点表現の定義により, n ビット精度の二変数数学関数は, $2n$ 入力 m 出力論理関数とみなすことができる. この多出力論理関数は, 二進ベクトルを整数とみなすことで, 整数関数に変換できる. 即ち, n ビット精度の二変数数学関数は, $B^{2n} \rightarrow P_m$ の $2n$ ビット精度の整数関数に変換できる. ただし, $P_m = \{0, 1, \dots, 2^m - 1\}$. 本稿では, 実数の数学関数を n ビット精度の固定小数点で表現することにより, 整数関数に変換する. 以下では, 特に断りがない限り, 数学関数は整数関数に変換されているものとする. また, 読みやすさのため X, Y の固定小数点表現の最下位ビットをそれぞれ x_0, y_0 とする.

例 1. 表 1(a)は, $f_r(X, Y) = \sqrt{X^2 + Y^2}$ の関数表を示す. この関数表を, 2 ビット精度の固定小数点で表現することにより, 表 1(b)の論理関数 $f_b(X, Y)$ を得る. そして, $f_b(X, Y)$ の二進ベクトルを整数と見なすことで, 表 1(c)の整数関数 $f(X, Y)$ を得る. 本稿で, 2 ビット精度の $\sqrt{X^2 + Y^2}$ は, 表 1(c)の整数関数 $f(X, Y)$ を意味する. (例終り)

表 1: 2 ビット精度の $f_r(X, Y) = \sqrt{X^2 + Y^2}$ の関数表

(a) f_r の関数表			(b) f_b の真理値表			(c) f の関数表		
X	Y	f_r	X	Y	f_b	X	Y	f
0.00	0.00	0.00	0.00	0.00	0.00	00	00	0
0.00	0.25	0.25	0.00	0.01	0.01	00	01	1
0.00	0.50	0.50	0.00	0.10	0.10	00	10	2
0.00	0.75	0.75	0.00	0.11	0.11	00	11	3
0.25	0.00	0.25	0.01	0.00	0.01	01	00	1
0.25	0.25	0.35	0.01	0.01	0.01	01	01	1
0.25	0.50	0.56	0.01	0.10	0.10	01	10	2
0.25	0.75	0.79	0.01	0.11	0.11	01	11	3
0.50	0.00	0.50	0.10	0.00	0.10	10	00	2
0.50	0.25	0.56	0.10	0.01	0.10	10	01	2
0.50	0.50	0.71	0.10	0.10	0.11	10	10	3
0.50	0.75	0.90	0.10	0.11	1.00	10	11	4
0.75	0.00	0.75	0.11	0.00	0.11	11	00	3
0.75	0.25	0.79	0.11	0.01	0.11	11	01	3
0.75	0.50	0.90	0.11	0.10	1.00	11	10	4
0.75	0.75	1.06	0.11	0.11	1.00	11	11	4

2.2 決定グラフ

本節では, 本稿で用いる決定グラフを簡単に定義する. 各決定グラフの定義や簡単化規則の詳細は, [6, 14, 24, 33, 34] を参照されたい.

定義 4. 二分決定グラフ(BDD: Binary Decision Diagram) [2, 17]は, 論理関数をグラフで表現したもので, 論理関数 f にシャノン展開 $f = \bar{x}_i f_0 + x_i f_1$ を繰り返し適用することで得られる. BDD は, 論理関数値 0 および 1 を表現する二つの終端節点と, 入力変数 x_i を表現する非終端節点から成る. 各非終端節点は, 変数値に対応する重みなしの二つの枝, 0-枝と 1-枝を持つ.

定義 5. MTBDD (Multi-Terminal BDD) [4]は, BDD を拡張した表現法であり, 整数関数を表現する. MTBDD には, 整数値を表現する複数の終端節点がある.

定義 6. BMD (Binary Moment Diagram) [3]は, 整数関数をグラフで表現したもので, 関数 f に算術展開 $f = f_0 + x_i(f_1 - f_0)$ を繰り返し適用することで得られる. BMD は, 算術係数を表現する終端節点と, 算術展開を表現する非終端節点から成る. 各非終端節点は, 算術展開の各項に対応する二つの枝を持つ.

定義 7. EVBDD (Edge-Valued BDD) [14, 15]は, BDD を拡張した表現法であり, 整数関数 f に展開 $f = \bar{x}_i f_0 + x_i(f_1' + \alpha)$ を繰り返し適用することで得られる. ここで, f_1' は, $f_1 = f(x_i = 1)$ から定数項 α を除いた関数である. EVBDD は, 0 を表現する一つの終端節点と, 重み付きの 1-枝を持った非終端節点

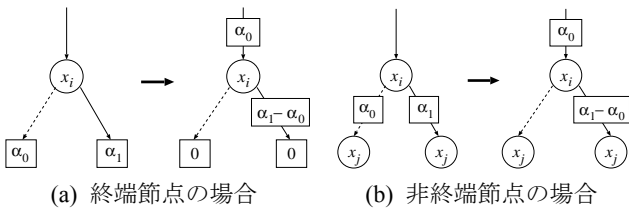


図 1. MTBDD から EVBDD への変換

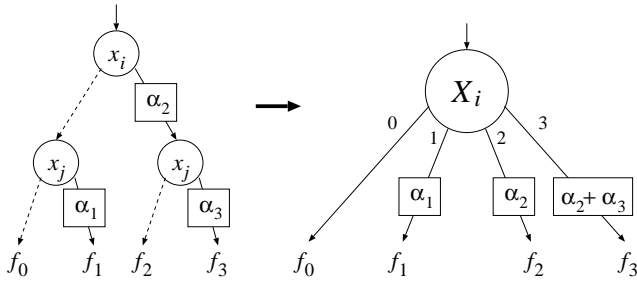


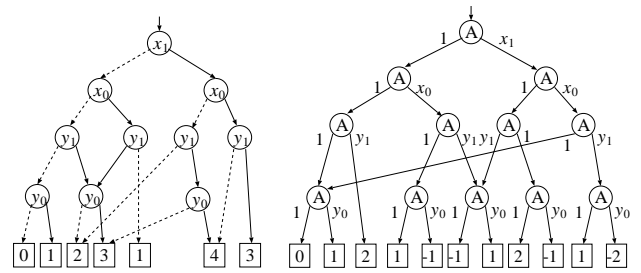
図 2. EVBDD から EVMDD への変換

から成る。ただし、枝の重みは整数値 α である。単純化された EVBDD では、各節点が表現する関数はすべて異なる。EVBDD は、MTBDD の各非終端節점에図 1 の変換を再帰的に適用することで生成できる。図中で、点線は 0-枝、実線は 1-枝を表している。

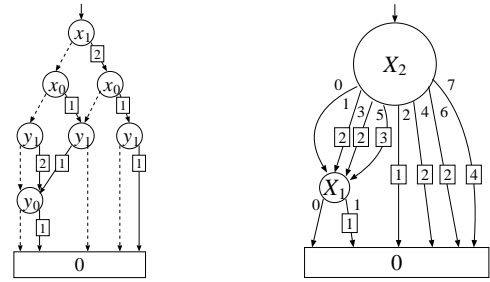
定義 8. n 個の二値変数の集合 $\{X\}$ において、 $\{X\} = \{X_u\} \cup \{X_{u-1}\} \cup \dots \cup \{X_1\}$ かつ $\{X_i\} \cap \{X_j\} = \phi$ ($i \neq j$) のとき、 $(X_u, X_{u-1}, \dots, X_1)$ を **変数 X の分割** といい、各 X_i のことを **超変数** と呼ぶ。ここで、 $|X_i| = k_i$ かつ $k_u + k_{u-1} + \dots + k_1 = n$ とする。各超変数 X_i を多値変数とみなすことで、整数関数 $f(X): B^n \rightarrow Z$ を **多値入力整数関数** $f(X_u, X_{u-1}, \dots, X_1): P_u \times P_{u-1} \times \dots \times P_1 \rightarrow Z$ に変換できる。ここで、 $P_i = \{0, 1, 2, \dots, 2^{k_i} - 1\}$ である。

定義 9. 多値決定グラフ (MDD: Multi-valued Decision Diagram) は、多値入力整数関数をグラフで表現したもので、多値入力整数関数にシャノン展開を繰り返し適用することで得られる[12]。MDD は、関数値を表現する終端節点と、多値変数を表現する非終端節点から成る。各非終端節点は、変数値に対応する重み無しの枝を持つ。各多値変数の定義域が異なってもよい関数を表現する MDD を、ヘテロジニアス MDD [19, 20] という。以下では、簡単のために、ヘテロジニアス MDD を単に MDD と略記する。

定義 10. EVMDD (Edge-Valued MDD) [21] は、MDD を拡張した表現法であり、多値入力整数関数を表現する。0 を表現する一つの終端節点と、重み付きの枝を持った非終端節点から成る。ただし、枝の重みは整数値であり、0-枝の重みは



(a) MTBDD (b) BMD



(c) EVBDD (d) EVMDD

図 3.2 ビット精度の $\sqrt{X^2 + Y^2}$ を表現する決定グラフ

0 である。EVMDD は、図 2 のように EVBDD の非終端節点を変数の分割に従って、グループ化することで得られる。

例 2. 図 3(a), (b), (c), (d) は、2 ビット精度の関数 $f(X, Y) = \sqrt{X^2 + Y^2}$ を表現する MTBDD, BMD, EVBDD, EVMDD をそれぞれ示している。図の見易さのため、共有していない終端節点があることに注意。図 3(a), (c) で、点線は 0-枝、実線は 1-枝を表し、図 3(b) の 'A' は、算術展開を表す。また、図 3(d) で、変数集合 $\{X\} \cup \{Y\}$ は、 $\{X_2\} = \{x_1, x_0, y_1\}$ 、 $\{X_1\} = \{y_0\}$ に分割されている。 $X = (10)_2$ 、 $Y = (10)_2$ に対する関数値 3 を得るために、MTBDD では、入力変数の値に従ってグラフを辿り、到達した終端節点の値から関数値を得る。BMD では、各非終端節点で算術展開を再帰的に計算することで関数値を得る。そして、EVBDD と EVMDD では、入力変数の値に従い、根節点から終端節点までグラフを辿ったときの枝の重みの総和から関数値を得る。ここで、EVMDD では、 $X_2 = 5$ 、 $X_1 = 0$ としてグラフを辿ることに注意。 (例終り)

3 二変数数学関数のグラフ表現

本節では、 l 限定 M_p 単調増大関数を定義し、 l 限定 M_p 単調増大関数を表現する EVBDD の節点数の上界を示す。実験により、二変数数学関数を表現する EVBDD は、MTBDD や BMD よりコンパクトであることを示す。

3.1 l 限定 M_p 単調増大関数

定義 11. 変数 X の 0 を含む定義域 I において、整数関数 $f(X): I \rightarrow Z$ が $0 \leq f(X+1) - f(X) \leq p$ かつ $f(0) = 0$ を満たすとき、 f は、定義域 I において **完全 M_p 単調増大**、または単に **M_p 単調増大関数** という。つまり、 M_p 単調増大関数 $f(X)$ とは、 $f(0) = 0$ でかつ、 X の値を 1 増やしたとき、 $f(X)$ の値が高々 p 増える関数である。

定義 12. n ビット精度整数関数 $f(X)$ において、 $l < n$ に対して、 l ビット精度のすべての部分関数 $g(X_l)$ が M_p 単調増大のとき、 f は l 限定 M_p 単調増大という。ただし、 $g(X_l) = f(a, X_l)$ 、 $\{X\} = \{x_{n-1}, x_{n-2}, \dots, x_0\}$ 、 $\{X_l\} = \{x_{l-1}, x_{l-2}, \dots, x_0\}$ 、 a は $(x_{n-1} x_{n-2} \dots x_l)_2$ へ割り当てられた値である。

定理 1. n ビット精度の整数関数 $f(X)$ が l 限定 M_p 単調増大であるとき、 $f(X)$ を表現する EVBDD の節点数は、高々

$$2^{n-l} + \sum_{i=1}^l (p+1)^{2^i-1} - l$$

である。ただし、 l は、 $2^{n-l} \geq (p+1)^{2^l-1}$ を満たす最大の整数であり、EVBDD の変数順序は、根節点から終端節点へ $x_{n-1}, x_{n-2}, \dots, x_0$ とする。

(証明) 付録参照。

定理 1 の l 限定 M_p 単調増大関数を表現する EVBDD の節点数の上界は、[21] で示した完全 M_p 単調増大関数における節点数の上界と同じである。

例 3. 16 ビット精度の l 限定 M_p 単調増大関数において、 $p = 1$ のとき、 $l = 3$ となり、定理 1 で与えられる節点数の上界は、8,327 になる。 $p = 3$ のときは、 $l = 2$ となり、節点数の上界は、16,450 になる。(例終り)

定義 13. n ビット精度整数関数 $f(X)$ において、 l ビット精度のすべての部分関数が M_p 単調増大関数 $g(X_l)$ または、 M_p 単調増大関数に定数を加えた関数 $g(X_l) + b$ であるとき、 f は **拡張 l 限定 M_p 単調増大** という。ただし、 b は整数であり、 $\{X\} = \{x_{n-1}, x_{n-2}, \dots, x_0\}$ 、 $\{X_l\} = \{x_{l-1}, x_{l-2}, \dots, x_0\}$ 、 $l < n$ である。

補題 1. $f(X)$ が拡張 l 限定 M_p 単調増大とする。 $1 \leq l' \leq l$ を満たす任意の l' において、 $f(X)$ は拡張 l' 限定 M_p 単調増大である。

(証明) 拡張 l 限定 M_p 単調増大関数の定義より、自明である。(証明終り)

表 2: 2 ビット精度二変数関数の関数表

(a) $\sqrt{X^2 + Y^2}$					(b) $X / (Y + 1)$					(c) $g(Z)$ の関数表				
Y	X				Y	X				$y_1 y_0$	$x_1 x_0$			
	0	1	2	3		0	1	2	3		00	01	10	11
0	0	1	2	3	0	0	1	2	3	00	0	-1	-2	-3
1	1	1	2	3	1	0	1	2	3	01	0	-1	-2	-2
2	2	2	3	4	2	0	1	1	2	10	0	-1	-1	-2
3	3	3	4	4	3	0	1	1	2	11	0	-1	-1	-2

補題 2. $f(X)$ が l 限定 M_p 単調増大とする。 f の l ビット精度部分関数に定数項を加えて得られる拡張 l 限定 M_p 単調増大関数を $g(X)$ とすると、 $g(X)$ と $f(X)$ は、同じ節点数の EVBDD で表現できる。

(証明) $f(X)$ を表現する EVBDD において、対応する部分関数を指している枝の重みに定数を加えることで、 $g(X)$ を表現できる。この変換は、枝の重みだけに影響するため、節点数に変化は生じない。(証明終り)

系 1. $f(X)$ を拡張 l 限定 M_p 単調増大関数とし、 $g(X)$ をそのアフィン変換 $g(X) = a f(X) + b$ とする。ここで、 a, b は整数である。このとき $g(X)$ と $f(X)$ は、同じ節点数の EVBDD で表現できる。

3.2 二変数数学関数

2 節で示したように、 n ビット精度の二変数関数は、 $2n$ ビット精度の整数関数に変換できる。つまり、 n ビット精度の二変数関数 $f(X, Y)$ は、

$$Z = 2^n X + Y = (x_{n-1} x_{n-2} \dots x_0 y_{n-1} y_{n-2} \dots y_0)_2$$

とすることで、 $2n$ ビット精度の一変数関数 $f(Z)$ に変換できる。 $f(Z)$ が、 $2^{2n-l} \geq (p+1)^{2^l-1}$ を満足する最大の整数 l において、 l 限定 M_p 単調増大であるとき、 $f(X, Y)$ を表現する EVBDD の節点数の上界は、定理 1 で与えられる。

例 4. 2 ビット精度の二変数関数 $\sqrt{X^2 + Y^2}$ は、表 2(a) に示すように、拡張 2 限定 M_1 単調増大関数 $f(Z)$ に変換できる。表 2(b) に示されている 2 ビット精度の二変数関数 $X / (Y + 1)$ は、表 2(c) の拡張 2 限定 M_1 単調増大関数 $g(Z)$ にアフィン変換 $-1 \times g(Z)$ を適用して得られる。(例終り)

以下では、EVBDD でコンパクトに表現できる二変数関数のクラスを解析的に求める。

補題 3. $h(Y)$ を n ビット精度の M_p 単調増大関数とする。 $h(Y)$ と任意の一変数整数関数 $g(X)$ の和で得られる二変数関数 $f(X, Y) = g(X) + h(Y)$ は、拡張 n 限定 M_p 単調増大関数になる。

(証明) 二変数関数 $f(X, Y)$ において、 X への値の割り当てで

得られる部分関数は、 $h(Y) + b$ で表現できる。ただし、 b は整数。よって、拡張 n 限定 M_p 単調増大関数の定義より、明らか。

(証明終り)

補題 4. $h(Y)$ を n ビット精度の M_p 単調増大関数とする。任意の一変数整数関数 $g(X)$ において、二変数関数 $f(X, Y) = g(X) - h(Y)$ は、拡張 n 限定 M_p 単調増大関数のアフィン変換で得られる。

(証明) 二変数関数 $f(X, Y)$ を、 $f(X, Y) = -(-g(X) + h(Y))$ と変形する。補題 3 より、 $-g(X) + h(Y)$ は、拡張 n 限定 M_p 単調増大になる。よって、補題が成立する。

(証明終り)

補題 5. $h(Y)$ を n ビット精度の M_p 単調増大関数、 $g(X)$ を $0 \leq g(X) \leq 1$ を満たす一変数実数関数とする。二つの関数の積で得られる n ビット精度の二変数関数 $f(X, Y) = g(X) h(Y)$ は、 n 限定 M_p 単調増大になる。

(証明) 二変数関数 $f(X, Y)$ において、 X への値の割り当てで得られる部分関数は、 $a h(Y)$ で表現できる。 a は、 $0 \leq a \leq 1$ を満たす実数なので、 $a h(Y) = 0$ と $0 \leq a h(Y+1) - a h(Y) \leq p$ を満たす。よって、 n 限定 M_p 単調増大関数の定義より、補題が成立する。

(証明終り)

補題 5 で、 $g(X)$ の値域が 1 より大きいとき、EVBDD の節点数は大きくなる場合がある。例えば、 n ビット乗算を表現する EVBDD は、少なくとも 2^n 個の節点が必要になる[33]。

補題 6. $h(Y)$ を n ビット精度の M_p 単調増大関数のアフィン変換、 $g(X)$ を $0 \leq g(X) \leq 1$ を満たす一変数実数関数とする。二つの関数の積で得られる n ビット精度の二変数関数 $f(X, Y) = g(X) h(Y)$ は、拡張 n 限定 M_p 単調増大関数のアフィン変換で得られる。

(証明) $h(Y) = a h_0(Y) + b$ とすると、

$$\begin{aligned} f(X, Y) &= g(X) (a h_0(Y) + b) \\ &= a (g(X) h_0(Y) + b g(X) / a) \end{aligned}$$

となる。補題 3 と補題 5 より、 $g(X) h_0(Y) + b g(X) / a$ は、拡張 n 限定 M_p 単調増大になる。よって、補題が成立する。

(証明終り)

例 5.2 ビット精度の $1 / (Y + 1)$ は、 M_1 単調増大関数のアフィン変換である[21]。例 4 で示したように、 $f(X, Y) = X / (Y + 1)$ は、拡張 2 限定 M_1 単調増大関数のアフィン変換で得られる。

(例終り)

以下では、実験により、補題で示した関数以外の二変数

表 3: 8 ビット精度二変数関数における決定グラフの節点数

数学関数	M_p	節点数			R_1	R_2
		MTBDD	BMD	EVBDD		
$\sqrt{X^2 + Y^2}$	M1	12,969	25,084	2,566	20	10
$\arctan(X / Y + 1)$	M1+	8,997	26,158	3,134	35	12
$\ln(X + 1) \sin(Y)$	M1	9,776	25,994	3,444	35	13
$\sqrt{X} \sin(Y)$	M1	11,543	26,542	3,483	30	13
$\sin(\sqrt{X^2 + Y^2})$	M1	11,521	27,858	4,013	35	14
$\sin(X Y)$	M1	11,282	21,746	3,789	34	17
$X / (Y + 1)$	M1+	9,664	25,878	3,162	33	12
$XY / \sqrt{X^2 + Y^2}$	M1	9,325	23,634	2,269	24	10
WaveRings	M3+	17,423	27,691	5,047	29	18
平均		11,389	25,621	3,434	30	13

関数の定義域: $0 \leq X < 1, 0 \leq Y < 1$

関数値の小数部は 8 ビット

M_{p+} : 拡張 8 限定 M_p 単調増大関数のアフィン変換

$R_1 = (\text{EVBDD}) / (\text{MTBDD}) \times 100, R_2 = (\text{EVBDD}) / (\text{BMD}) \times 100$

決定グラフの変数順序は、sifting アルゴリズム[23]で生成した。

数学関数も拡張 l 限定 M_p 単調増大関数に変換できることを示す。表 3 は、8 ビット精度の二変数数学関数(入力変数と関数値の小数部が、ともに 8 ビット)を表現する決定グラフの節点数を比較している。表 3 の二変数関数は、微分積分学の本[1, 11, 13]から選択した。表中で“ M_p ”の列は、数学関数が属する拡張 8 限定 M_p 単調増大関数のクラスを示し、 M_{p+} は、数学関数が拡張 8 限定 M_p 単調増大関数のアフィン変換によって得られることを表している。表中の WaveRings は、以下で定義される関数である[11]。

$$\text{WaveRings} = \frac{\cos(\sqrt{X^2 + Y^2})}{\sqrt{X^2 + Y^2 + 0.25}}$$

与えられた定義域において緩やかに変化する二変数数学関数は、 p の値が小さい拡張 l 限定 M_p 単調増大関数に変換できる。そのため、そのような二変数数学関数は、定理 1 で示したように、コンパクトな EVBDD で表現できる。実際に、表 3 から、本実験で用いた数学関数は、拡張 8 限定 M_1 または M_3 単調増大関数に変換でき、MTBDD や BMD よりも遥かに少ない節点数の EVBDD で表現できることがわかる。EVBDD の非終端節点は、重み付きの枝を持っているので、非終端節点一つあたりに必要なメモリ量は、MTBDD や BMD の非終端節点よりもわずかに大きくなるが、EVBDD の節点数が MTBDD や BMD に比べ十分少ないので、枝の重みによるメモリ量の増加を考慮しても EVBDD の方がコンパクトな表現法であるといえる[21]。以上の結果から、上記の補題で解析した二変数数学関数以外の様々な関数においても、 p の値が小さい拡張 l 限定 M_p 単調増大関数に変換でき、EVBDD でコンパクトに表現できることがわかる。

文献[21]で示したように、EVBDD を多値決定グラフ(EVMDD)に拡張することで、決定グラフのメモリ量やパス長をさらに削減できる。次の節で、EVMDD に基づく数値計算回路の構成とともに、EVMDD の有用性を示す。

表 5: 8 ビット精度二変数関数の FPGA 実装結果

FPGA デバイス: Altera Stratix EP1S25F672C8												
論理合成ツール: Altera QuartusII 7.1 (速度最適化, タイミング制約: 100MHz)												
数学関数	EVBDD					EVMDD					比率 [%]	
	LE	メモリ量 [ビット]	周波数 [MHz]	LPL	遅延 [nsec.]	LE	メモリ量 [ビット]	周波数 [MHz]	LPL	遅延 [nsec.]	メモリ	遅延
$\sqrt{X^2 + Y^2}$	367	155,736	80	16	201	96	103,080	66	5	76	66	38
$\arctan(X / Y + 1)$	256	132,174	79	16	204	94	88,760	66	6	91	67	44
$\ln(X + 1) \sin(Y)$	169	142,640	79	16	202	91	86,256	66	5	76	60	37
$\sqrt{X} \sin(Y)$	317	157,080	81	16	198	91	91,404	67	5	74	58	38
$\sin(\sqrt{X^2 + Y^2})$	204	184,968	69	16	231	93	101,916	65	5	77	55	33
$\sin(X Y)$	365	151,520	81	16	197	91	90,828	67	5	74	60	38
$X / (Y + 1)$	338	145,782	82	16	195	91	88,236	67	5	75	61	38
$XY / \sqrt{X^2 + Y^2}$	269	145,200	80	16	200	94	99,826	67	5	75	69	37
<i>WaveRings</i>	114	235,934	68	16	234	101	144,892	65	5	77	61	33
平均	267	161,226	78	16	207	94	99,466	66	5	77	62	37

LE: 論理素子数 周波数: 最大動作周波数 LPL: 決定グラフの最長パス長
 遅延: 最大遅延時間 = LPL / 周波数 メモリの比率 = EVMDD のメモリ量 / EVBDD のメモリ量 × 100
 遅延の比率 = EVMDD の遅延 / EVBDD の遅延 × 100

$$\text{シフトデータ} = \sum_{j=1}^{u-1} k_j$$

$$\text{マスクのビット数} = \max_{1 \leq j \leq u} (k_j)$$

$$\text{マスクの値} = 2^{k_i} - 1$$

となる。ただし、EVMDD の変数順序は、根から X_u, X_{u-1}, \dots, X_1 とし、 $k_j = |X_j|$ ($j = 1, 2, \dots, u$) とする。

本数値計算回路のハードウェア量や性能は、EVMDD のサイズやパス長に大きく依存する。そのため、MDD のメモリ量最小化アルゴリズムや平均パス長最小化アルゴリズム [19, 20] は、数値計算回路のメモリ量削減や高速化に有効である。

4.3 FPGA 実装結果

表 5 は、表 3 の二変数数学関数を図 4 の回路で FPGA 実装した結果を示している。表中の“EVBDD”の列は、EVBDD から生成した数値計算回路の結果を示し、“EVMDD”の列は EVMDD から生成した数値計算回路の結果を示している。

EVBDD から生成した数値計算回路では、アドレス計算回路内の AND ゲートと加算器が不要になるため、EVMDD から生成した数値計算回路より動作周波数を高くできる。しかし、EVMDD のパス長は、EVBDD より大幅に短いため、関数値を得るまでの時間は EVMDD の方が短くなる。EVMDD から生成された数値計算回路において、関数値を得るために必要となる最大遅延時間は、平均して、EVBDD で必要な最大遅延時間の 37% になる。メモリ量に関しても、EVMDD は、平均して、EVBDD の 62% だけを必要とする。

以上の結果から、EVMDD は、二変数数学関数の高速かつコンパクトな数値計算回路の設計に有用であることがわかる。

5 結論とコメント

本稿では、 l 限定 M_p 単調増大関数という関数のクラスを定義し、 l 限定 M_p 単調増大関数を表現する EVBDD の節点数の上界を示した。そして、 l 限定 M_p 単調増大関数とそのアフィン変換で得られる関数を表現する EVBDD は、 p が小さいとき、他の表現法 (MTBDD, BMD など) に比べコンパクトであることを理論的に示した。また、EVMDD に基づく数値計算回路の設計法を提案し、EVMDD を用いることで正確な数値計算回路を高速かつコンパクトに実現できることを示した。いくつかの二変数数学関数を FPGA に実装した結果、EVMDD に基づく数値計算回路に必要なメモリ量と遅延時間は、平均して、EVBDD に必要なメモリ量と遅延時間の 62% と 37% になることを示した。

謝辞

本研究の一部は、平成 20 年度科学研究費補助金 (若手研究 (B)) 課題番号 200700051 および平成 20 年度広島市立大学教員特定研究費 (一般研究) 課題番号 8108 による。

参考文献

- [1] H. Anton, *Multivariable Calculus*, John Wiley & Sons, Inc., 1995.
- [2] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. on Computers*, Vol. C-35, No. 8, pp. 677-691, Aug. 1986.
- [3] R. E. Bryant and Y-A. Chen, "Verification of arithmetic circuits with binary moment diagrams," *Design Automation Conference*, pp. 535-541, 1995.
- [4] E. M. Clarke, K. L. McMillan, X. Zhao, M. Fujita, and J. Yang, "Spectral transforms for large Boolean functions with applications to technology mapping," *Design Automation Conference*, pp. 54-60, June 1993.

- [5] J. Detrey and F. de Dinechin, "Table-based polynomials for fast hardware function evaluation," *Inter. Conf. on Application-Specific Systems, Architectures, and Processors (ASAP)*, pp. 328-333, 2005.
- [6] R. Drechsler and B. Becker, *Binary Decision Diagrams: Theory and Implementation*, Kluwer Academic Publishers, 1998.
- [7] R. Gutierrez and J. Valls, "Implementation on FPGA of a LUT based atan(y/x) operator suitable for synchronization algorithms," *International Conference on Field Programmable Logic and Applications*, pp. 472-475, Aug. 2007.
- [8] Z. Huang and M. D. Ercegovic, "FPGA implementation of pipelined on-line scheme for 3-D vector normalization," *International Symposium on Field-Programmable Custom Computing Machines*, pp. 61-70, Apr. 2001.
- [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)*, pp. 73-76, Jan. 2000.
- [10] Y. Iguchi, T. Sasao, and M. Matsuura, "Evaluation of multiple-output logic functions using decision diagrams," *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 312-315, Jan. 2003.
- [11] 井上 真, 見る微分積分学: *Mathematica* によるイメージレーニング, 東京電機大学出版局, 1997.
- [12] T. Kam, T. Villa, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "Multi-valued decision diagrams: Theory and applications," *Multiple-Valued Logic: An International Journal*, Vol. 4, No. 1-2, pp. 9-62, 1998.
- [13] 小林 道正, *Mathematica* による関数グラフィックス, 森北出版, 1997.
- [14] Y-T. Lai and S. Sastry, "Edge-valued binary decision diagrams for multi-level hierarchical verification," *Design Automation Conference*, pp. 608-613, 1992.
- [15] Y-T. Lai, M. Pedram, and S. B. Vrudhula, "EVBDD-based algorithms for linear integer programming, spectral transformation and functional decomposition," *IEEE Trans. on Comput.-Aided Des. Integr. Circuits Syst.*, Vol. 13, No. 8, pp. 959-975, Aug. 1994.
- [16] D.-U. Lee, W. Luk, J. Villasenor, and P. Y. K. Cheung, "Hierarchical segmentation schemes for function evaluation," *International Conference on Field Programmable Technology*, pp. 92-99, 2003.
- [17] C. Meinel and T. Theobald, *Algorithms and Data Structures in VLSI Design: OBDD – Foundations and Applications*, Springer, 1998.
- [18] J.-M. Muller, *Elementary Functions: Algorithms and Implementation*, Birkhauser Boston, Inc., Secaucus, NJ, 1997.
- [19] S. Nagayama and T. Sasao, "Compact representations of logic functions using heterogeneous MDDs," *IEICE Trans. on Fundamentals*, Vol. E86-A, No. 12, pp. 3168-3175, Dec. 2003.
- [20] S. Nagayama and T. Sasao, "On the optimization of heterogeneous MDDs," *IEEE Trans. on Comput.-Aided Des. Integr. Circuits Syst.*, Vol. 24, No. 11, pp. 1645-1659, Nov. 2005.
- [21] S. Nagayama and T. Sasao, "Representations of elementary functions using edge-valued MDDs," *International Symposium on Multiple-Valued Logic*, 2007.
- [22] J.-A. Piñeiro, S. F. Oberman, J.-M. Muller, and J. D. Bruguera, "High-speed function approximation using a minimax quadratic interpolator," *IEEE Trans. on Computers*, Vol. 54, No. 3, pp. 304-318, Mar. 2005.
- [23] R. Rudell, "Dynamic variable ordering for ordered binary decision diagrams," *International Conference on Computer-Aided Design*, pp. 42-47, Nov. 1993.
- [24] T. Sasao and M. Fujita (eds.), *Representations of Discrete Functions*, Kluwer Academic Publishers, 1996.
- [25] T. Sasao, *Switching Theory for Logic Synthesis*, Kluwer Academic Publishers, 1999.
- [26] T. Sasao and S. Nagayama, "Representations of elementary functions using binary moment diagrams," *International Symposium on Multiple-Valued Logic*, 2006.
- [27] T. Sasao, S. Nagayama, and J. T. Butler, "Numerical function generators using LUT cascades," *IEEE Trans. on Computers*, Vol. 56, No. 6, pp. 826-838, Jun. 2007.
- [28] M. J. Schulte and J. E. Stine, "Approximating elementary functions with symmetric bipartite tables," *IEEE Trans. on Computers*, Vol. 48, No. 8, pp. 842-847, Aug. 1999.
- [29] R. Stankovic and J. Astola, *Spectral Interpretation of Decision Diagrams*, Springer Verlag, New York, 2003.
- [30] R. Stankovic and J. Astola, "Remarks on the complexity of arithmetic representations of elementary functions for circuit design," *Workshop on Applications of the Reed-Muller Expansion in Circuit Design and Representations and Methodology of Future Computing Technology*, pp. 5-11, 2007.
- [31] N. Takagi and S. Kuwahara, "A VLSI algorithm for computing the Euclidean norm of a 3D vector," *IEEE Trans. on Computers*, Vol. 49, No. 10, pp. 1074-1082, Oct. 2000.
- [32] M. A. Thornton, R. Drechsler, and D. M. Muller, *Spectral Techniques in VLSI CAD*, Springer, 2001.
- [33] I. Wegener, *Branching Programs and Binary Decision Diagrams: Theory and Applications*, SIAM, 2000.
- [34] S. N. Yanushkevich, D. M. Muller, V. P. Shmerko, and R. S. Stankovic, *Decision Diagram Techniques for Micro- and Nanoelectronic Design*, CRC Press, Taylor & Francis Group, 2006.

付録

定理 1 を証明するために、以下の二つの補題を用いる。

補題 A. [21] l ビット精度の M_p 単調増大関数は、高々 $(p + 1)^{2^{l-1}}$ 個存在する。

定義 A. 整数関数を表現する EVBDD の集合において、等価な部分グラフを共有し、単純化して表現したとき、複数の根節点を持つ EVBDD を SEVBDD (Shared EVBDD) という。

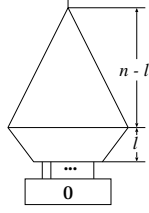


図 A. EVBDD の分割

補題 B. [21] l ビット精度のすべての M_p 単調増大関数を同時に表現する SEVBDD の非終端節点数を $\eta(l, p)$ とすると,

$$\eta(l, p) = \sum_{i=1}^l (p+1)^{2^{i-1}} - l$$

が成立する. ただし, SEVBDD の変数順序は, 根節点から終端節点へ $x_{l-1}, x_{l-2}, \dots, x_0$ とする.

定理 1 の証明: 図 A のように, 関数 f を表現する EVBDD を上下に二分割する. このとき, 下部で, すべての l ビット精度の M_p 単調増大関数を表現し, 上部では, それらから一つを選択する関数を表現するものとする. 上部を表現する EVBDD は, 完全二分木するとき節点数が上限に達し, その節点数は,

$$2^{n-l} - 1 \quad (\text{i})$$

である. 下部では, すべての l ビット精度の M_p 単調増大関数が表現されたとき, EVBDD の節点数が最大となる. 補題 B より, その節点数は,

$$\eta(l, p) = \sum_{i=1}^l (p+1)^{2^{i-1}} - l \quad (\text{ii})$$

である. (i), (ii) より, 関数 f を表現する EVBDD の非終端節点数は, 高々

$$2^{n-l} + \sum_{i=1}^l (p+1)^{2^{i-1}} - l - 1$$

になる. これに終端節点数 1 を加えることで定理を得る. 補題 A より, EVBDD の下部で表現可能な M_p 単調増大関数の個数は, $(p+1)^{2^{l-1}}$ であり, それは上部で選択可能な関数の数 2^{n-l} を超えないため, 関係

$$2^{n-l} \geq (p+1)^{2^{l-1}}$$

を得る.

(証明終り)