

多値決定グラフを用いた単調初等関数の浮動小数点表現 Floating-Point Representations of Monotone Elementary Functions Using MDDs

永山 忍¹ 笹尾 勤² ジョンバトラー³
Shinobu Nagayama¹ Tsutomu Sasao² Jon T. Butler³

- ¹ 広島市立大学 情報工学科, 〒 731-3194 広島市 安佐南区 大塚東 3-4-1
Department of Computer and Network Engineering, Hiroshima City University
- ² 九州工業大学 電子情報工学科, 〒 820-8502 福岡県 飯塚市 川津 680-4
Department of Computer Science and Electronics, Kyushu Institute of Technology
- ³ 海軍大学院大学 電気情報工学科, アメリカ カリフォルニア州 モントレー
Department of Electrical and Computer Engineering, Naval Postgraduate School

概要

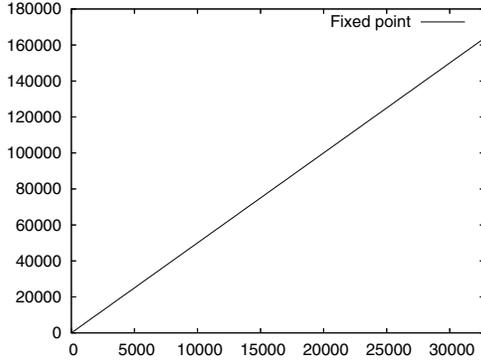
This paper proposes a design method for floating-point numerical function generators (NFGs) using multi-valued decision diagrams (MDDs) for monotone elementary functions. We convert such real valued functions into integer functions by the floating-point representation, and represent the integer functions using edge-valued MDDs (EVMDDs). To show that EVMDDs can compactly represent the integer functions converted from elementary functions by the floating-point representation, this paper introduces a transition point and a new class of integer functions, an Mp -monotone increasing function with transition points, and derives an upper bound on the number of nodes in an edge-valued binary decision diagram (EVBDD) for the Mp -monotone increasing function with transition points. It shows that EVBDDs represent Mp -monotone increasing functions with transition points compactly when the number of transition points and p are small. Experimental results show that many monotone elementary functions can be converted into Mp -monotone increasing functions with a small number of transition points and a small p , and can be represented by EVBDDs with fewer nodes by one or two orders of magnitude than MTBDDs and BMDs. Since EVMDDs have shorter paths and smaller memory size than EVBDDs, EVMDDs can produce fast and compact floating-point NFGs for elementary functions. FPGA implementation results of our floating-point NFG show the usefulness of the proposed method.

1 はじめに

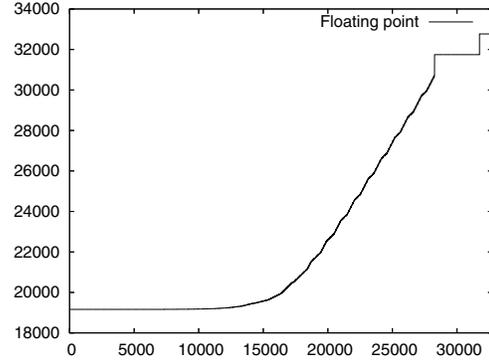
初等関数は、加算や乗算と同様に基本的な演算として、コンピュータグラフィックスやデジタル信号処理などの様々な分野で広く利用されており、初等関数を計算する様々な回路の設計法が提案されている [16]. 高速な初等関数回路を設計するために、多くの場合 [4, 13, 20, 24, 26], 固定小数点表現を用いるが、固定小数点表現は、高速な回路をもたらす一方で、関数の定義域や値域が広がると数値を表現するために必要なビット数が増加し、回路規模が大きくなるという欠点を持つ。ビット数を過度に増加させずに広い定義域や値域を表現する方法として、浮動小数点表現がしばしば利用され、IEEE の標準規格 [8] としても採用されているが、浮動小数点表現を用いると回路が複雑になり、設計によっては、十分な性能を達成できない場合がある。実際、浮動小数点での高速な初等関数回路の設計は難しく、一部の初等関数しか設計法が知

られていない [5, 6, 9, 25, 30]. 浮動小数点初等関数回路の既存設計法の多くは、特定の関数だけを対象とした専用回路の設計法なので、関数毎に異なる設計法と回路構成が必要になり、未知の関数に必ずしも適用することができない。本研究では、この問題点を解決するために、多様な初等関数を実現できる組織的な設計法を提案する。

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 [7, 14], 様々な決定グラフを用いたシステムチックな設計法が確立されている [15, 21, 29, 31]. そこで、本研究では、決定グラフを用いた初等関数回路の設計法について考える。決定グラフを用いた設計法では、与えられた関数に適した決定グラフを用いることが重要である。しかし、私たちの知る限り、浮動小数点表現による初等関数のグラフ表現法に関する研究は、報告されておらず、浮動小数点表現の初等関数に適した決定グラフは、未だ知られていない。文献 [18, 19, 23, 27] で、固定小数点表現の



(a) 固定小数点表現で得られた整数関数.
整数ビット: 15, 小数ビット: 0
定義域: $0 \leq X \leq 32,767$
正の最小値: 1



(b) 浮動小数点表現で得られた整数関数.
指数部: 5 ビット, 仮数部: 10 ビット
定義域: $0 \leq X < \infty$ (最大値: 131,040)
正の最小値: 2^{-24}

図 1: $f(X) = 5X + 13.7$ の固定小数点表現と浮動小数点表現.

初等関数に適した決定グラフが報告されているが、図 1 に示すように、固定小数点表現と浮動小数点表現では、同じ初等関数でも、全く異なる整数関数に変換される。図 1 は、線形関数 $f(X) = 5X + 13.7$ を、固定小数点と浮動小数点により変換して得られた整数関数を示している。このように単純な初等関数でも、浮動小数点表現では、複雑な整数関数に変換されてしまうため、固定小数点表現とは、別のクラスの関数として扱う必要がある。

そこで本研究では、転位点と、それを含んだ Mp 単調増大関数という関数のクラスを定義し、このクラスの関数を表現する決定グラフの複雑さを解析する。理論的解析と実験結果により、EVMDD (Edge-Valued Multi-valued Decision Diagram) は、固定小数点表現だけでなく浮動小数点表現の初等関数もコンパクトに表現できることを示す。また、EVMDD を用いることで、多様な初等関数のコンパクトかつ高速な浮動小数点回路を容易に設計できることを示す。

本稿は、以下のように構成されている。第 2 節では、実数 (初等) 関数を整数関数へ変換するための浮動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、転位点と、それを含んだ Mp 単調増大関数を定義し、転位点を含んだ Mp 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。そして、多くの単調初等関数は、転位点を含んだ Mp 単調増大関数に変換でき、EVMDD でコンパクトに表現できることを示す。最後に、第 4 節で、EVMDD に基づく浮動小数点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 $Z \rightarrow Z$, そして, 実数関数は写像 $R \rightarrow R$ である.

定義 2 n ビット精度の二進浮動小数点で表現された数値 X を

$$X = (s, e_{a-1}, e_{a-2}, \dots, e_0, d_{b-1}, d_{b-2}, \dots, d_0)_2$$

と表記する. この表記は, 符号ビット $s \in B$, 指数部 $E = (e_{a-1}, e_{a-2}, \dots, e_0)_2$, $e_i \in B$, および仮数部 $D = (d_{b-1}, d_{b-2}, \dots, d_0)_2$, $d_i \in B$ から成る. ただし, $B = \{0, 1\}$, a, b はそれぞれ指数部と仮数部のビット数であり, $n = a + b + 1$ である. このとき X は, 指数部 E の値によって, 表 1 のような値をとる. $|X| < 2^{-2^{a-1}}$ のとき, X は非正規化数として表現される. このとき, 指数部 E は, $E_s = 2^{a-1} - 2$ でバイアスされ, 仮数部 D は, 1 未満の数値の小数部のみを表現する. $2^a \leq |X|$ のとき, X は無限大として表現され, X を定義できないとき, X は非数として表現される. それ以外の X は, 正規化数として表現される. このとき, 指数部 E は, $E_n = 2^{a-1} - 1$ でバイアスされ, 仮数部 D は, 1 以上 2 未満の数値の小数部のみを表現する.

表 1: 指数部 a ビット, 仮数部 b ビットの浮動小数点数 X の値.

クラス	指数部 E	仮数部 D	X の値
ゼロ	$(0, 0, \dots, 0)_2$	$(0, 0, \dots, 0)_2$	0
非正規化数	$(0, 0, \dots, 0)_2$	$\neq (0, 0, \dots, 0)_2$	$(-1)^s \times 2^{-E_s} \times 0.D$
無限大	$(1, 1, \dots, 1)_2$	$(0, 0, \dots, 0)_2$	$(-1)^s \times \infty$
非数	$(1, 1, \dots, 1)_2$	$\neq (0, 0, \dots, 0)_2$	NaN
正規化数	上記以外		$(-1)^s \times 2^{E-E_n} \times 1.D$

非正規化数のバイアス値: $E_s = 2^{a-1} - 2$

正規化数のバイアス値: $E_n = 2^{a-1} - 1$

表 2: 8 ビット精度 (指数部:3 ビット, 仮数部:4 ビット) の \sqrt{X} の関数表

(a) \sqrt{X} の関数表.

X	\sqrt{X}
0.000000	0.000000
0.015625	0.125000
0.031250	0.171875
0.046875	0.218750
0.062500	0.250000
0.078125	0.281250
0.093750	0.312500
0.109375	0.328125
⋮	⋮

(b) 論理関数 $f_b(X)$ の真理値表.

X	$f_b(X)$
00000000	00000000
00000001	00001000
00000010	00001011
00000011	00001110
00000100	00010000
00000101	00010010
00000110	00010100
00000111	00010101
⋮	⋮

(c) 整数関数 $f(X)$ の関数表.

X	$f(X)$
0	0
1	8
2	11
3	14
4	16
5	18
6	20
7	21
⋮	⋮

IEEE 標準規格 [8] では, $a = 5, b = 10$ の半精度, $a = 8, b = 23$ の単精度, $a = 11, b = 52$ の倍精度, そして, $a = 15, b = 112$ の 4 倍精度が規定されている.

n ビット精度の浮動小数点表現により, 実数関数を, n 入力 n 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, 実数関数は, n ビット精度の浮動小数点表現により, n ビットの整数関数 $P_n \rightarrow P_n$ に変換できる. ただし, $P_n = \{0, 1, \dots, 2^n - 1\}$. 本稿では, 実数の初等関数を n ビット精度の浮動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, 初等関数は整数関数に変換されているものとする. また, 読み易さのため X の浮動小数点表現における各ビットを x_i とし, 最下位ビットを x_0 とする.

例 1 表 2(a) は, \sqrt{X} の関数表を示す. この関数表を, 8 ビット精度 (指数部:3 ビット, 仮数部:4 ビット) の浮動小数点で表現することにより, 表 2(b) の論理関数 $f_b(X)$ を得る. そして, $f_b(X)$ の二進ベクトルを整数と見なすこと

で, 表 2(c) の整数関数 $f(X)$ を得る. 本稿で, 8 ビット精度の \sqrt{X} は, 表 2(c) の整数関数 $f(X)$ を意味する. (例終り)

2.2 決定グラフ (Decision Diagram)

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, [7, 21, 29, 31] を参照されたい.

定義 3 二分決定グラフ (*BDD: Binary Decision Diagram*) [1, 14] は, 論理関数をグラフで表現したもので, 論理関数にシャノン展開 $f = \bar{x}_i f_0 + x_i f_1$ 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 4 *MTBDD (Multi-Terminal BDD)* [3] 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.

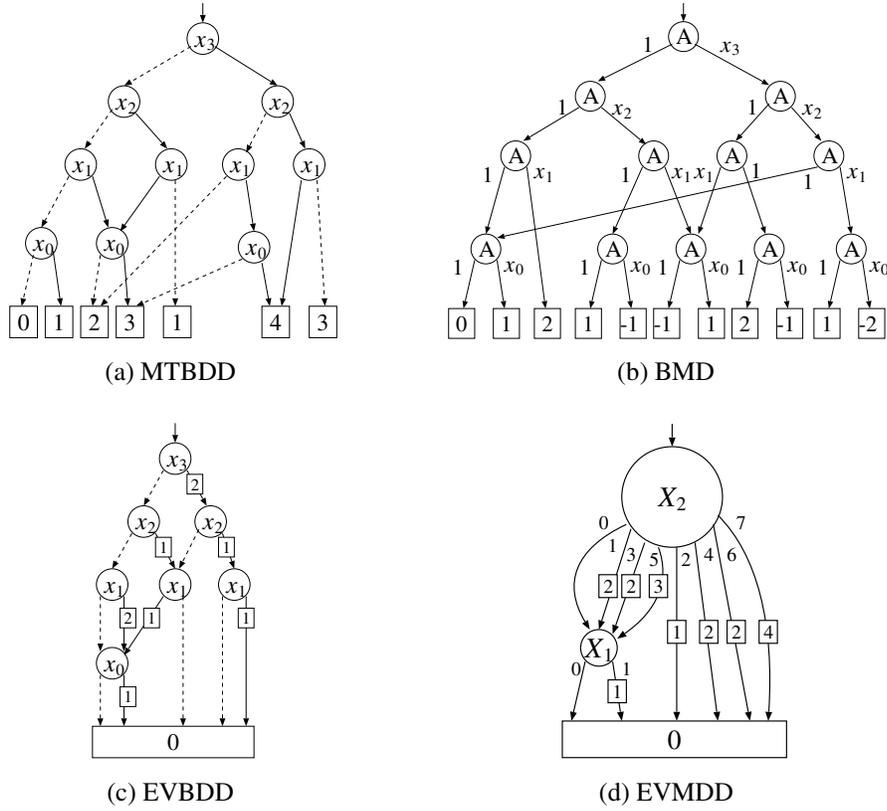


図 2: 4 ビット整数関数を表現する 4 種類の決定グラフ

定義 5 BMD (Binary Moment Diagram) [2] は、整数関数をグラフで表現したもので、整数関数に算術展開 $f = f_0 + x_i(f_1 - f_0)$ 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 6 EVBDD (Edge-Valued BDD) [11, 12] は、BDD を拡張した整数関数の表現法であり、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。

定義 7 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 $\{X_i\} \cap \{X_j\} = \emptyset$ ($i \neq j$) 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 $f(X) : Z \rightarrow Z$ 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 8 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 [10]。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 [17] という。以下では、簡単のため、ヘテロジニアス MDD を単に MDD と略記する。

定義 9 EVMDD (Edge-Valued MDD) [18] 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 0 である。

例 2 図 2(a), (b), (c), (d) は、4 ビット整数関数を表現する MTBDD, BMD, EVBDD, EVMDD をそれぞれ示している。

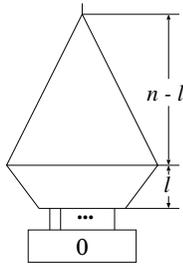


図 3: EVBDD の分割.

図の見やすさのため、共有していない終端節点があることに注意. 図 2(a), (c) で, 点線は 0-枝, 実線は 1-枝を表し, 図 2(b) の 'A' は, 算術展開を表す. また, 図 2(d) で, 変数集合 $\{X\} \cup \{Y\}$ は, $\{X_2\} = \{x_3, x_2, x_1\}$, $\{X_1\} = \{x_0\}$ に分割されている. $X = (1, 0, 1, 0)_2$ 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 浮動小数点初等関数のグラフ表現

本節では, 転位点と, それを含んだ M_p 単調増大関数を定義し, 転位点を含んだ M_p 単調増大関数を表現する EVBDD の節点数の上限を示す. 実験により, 浮動小数点初等関数を表現する EVBDD は, *MTBDD* や *BMD* よりコンパクトであることを示す.

3.1 転位点と M_p 単調増大関数

定義 10 変数 X の 0 を含む定義域 $I \subset Z$ 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, $f(0) = 0$ であつ, 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 11 ある整数 p が与えられたとき, 整数関数 $f(X) : Z \rightarrow Z$ において, $f(T+1) - f(T) < 0$ または $p < f(T+1) - f(T)$ を満足する点 T を, **転位点** とよぶ.

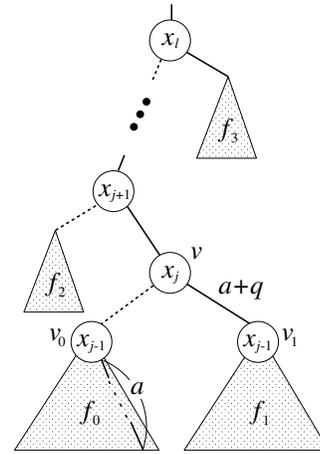


図 4: 転位点を表現する EVBDD.

定義 12 ある整数 p が与えられたとき, 整数関数 $f(X) : I \rightarrow Z$ で, k 個の転位点を除く全ての X において $0 \leq f(X+1) - f(X) \leq p$ かつ $f(0) = 0$ を満たすとき, f は, 定義域 I において k 個の転位点を含む M_p 単調増大関数という. つまり, k 個の転位点を含む M_p 単調増大関数 $f(X)$ とは, $f(0) = 0$ であつ, 転位点以外の X において, 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.

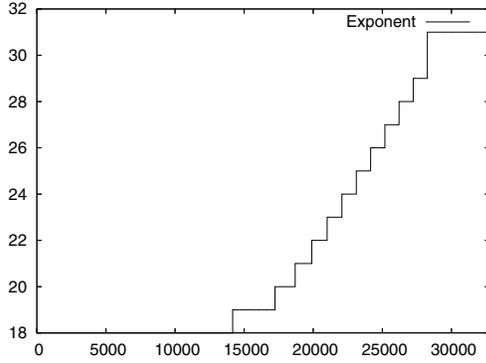
定理 1 n ビットの整数関数 $f(X)$ が k 個の転位点を含む M_p 単調増大関数であるとき, $f(X)$ を表現する EVBDD の節点数は, 高々

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

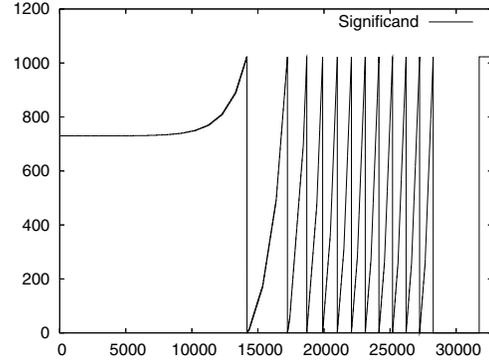
である. ただし, l は, $2^{n-l} \geq (p+1)^{2^l-1} + k$ を満たす最大の整数であり, *EVBDD* の変数順序は, 根節点から終端節点へ $x_{n-1}, x_{n-2}, \dots, x_0$ とする.

(証明) 図 3 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 単調増大関数と転位点を表現し, 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, $2^{n-l} - 1$ である. 下部では, すべての l ビット精度の M_p 単調増大関数と転位点が表現されたとき, EVBDD の節点数が最大となる. 文献 [18, 19] より, すべての l ビット精度の M_p 単調増大関数を表現するために必要な節点数は,

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



(a) 指数部.



(b) 仮数部.

図 5: $f(X) = 5X + 13.7$ ($0 \leq X < \infty$) の半精度の浮動小数点表現.

なので、以下では、下部で転位点を表現するために必要な節点数の上界を示す。

まず、1 個の転位点を含んでいる場合を考える。転位点 T を

$$T = (t_{n-1}, t_{n-2}, \dots, t_{j+1}, t_j, t_{j-1}, \dots, t_0)_2$$

とすると、ある整数 j において、

$$T = (t_{n-1}, t_{n-2}, \dots, t_{j+1}, 0, 1, 1, \dots, 1)_2$$

$$T + 1 = (t_{n-1}, t_{n-2}, \dots, t_{j+1}, 1, 0, 0, \dots, 0)_2$$

が成立する。 $q = f(T+1) - f(T)$ とすると、 $q < 0$ または $p < q$ となり、図 4 のような EVBDD で転位点を表現できる。この図で、 a は、節点 v_0 から辿った 1-枝の重みの和であり、部分関数 f_0, f_1, f_2, f_3 は、 M_p 単調増大関数である。図に示されているように、節点 v およびその親節点の入次数は、1 であり、節点が数珠繋ぎになっている*。この数珠繋ぎになった節点以外は、 M_p 単調増大関数を表現しているため、これらの節点だけが、転位点を表現するために必要な節点となる。 $j = 0$ のとき、この節点数が最大となるため、1 個の転位点を表現するのに、高々 l 個の節点が必要になる。

k 個の転位点が、それぞれ別々に表現されたとき、EVBDD の下部の節点数は最大になり、その節点数は、

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

となる。これに上部の節点数 $2^{n-l} - 1$ と終端節点数 1 を加えることで、定理が得られる。EVBDD の下部で表

* 入次数 2 以上の節点があったとすると、転位点の数が 1 であることに反する。

現可能な関数の数は、すべての M_p 単調増大関数の個数 $(p+1)^{2^{l-1}}$ と転位点の数 k であり、この数は、上部で選択可能な関数の数 2^{n-l} を超えないため、関係

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

を得る。

(証明終)

定理 1 から、転位点数 k の増加に対して、EVBDD の節点数は、 M_p 単調増大関数を表現している EVBDD の節点数に比べ、高々 kl 個しか増加しないことがわかる。すなわち、 M_p 単調増大関数は、転位点を含んでもほぼ同程度の複雑さ(大きさ)の EVBDD で表現できる。

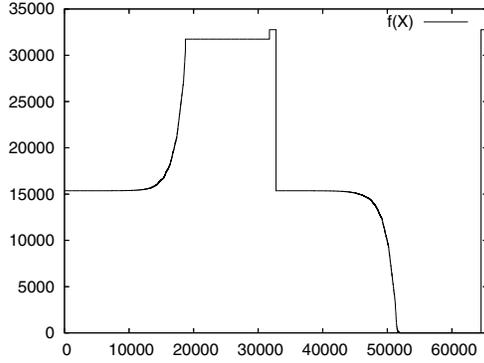
系 1 $f(X)$ を k 個の転位点を含む M_p 単調増大関数とし、 $g(X)$ をそのアフィン変換

$$g(X) = af(X) + b$$

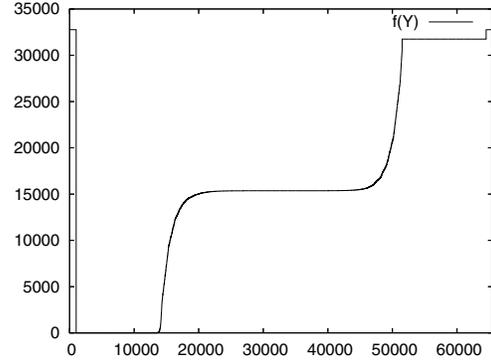
とする。ここで、 a, b は整数である。このとき $g(X)$ と $f(X)$ は、同じ節点数の EVBDD で表現できる。

3.2 浮動小数点初等関数から M_p 単調増大関数への変換

図 1 で示したように、固定小数点表現とは違い、初等関数の浮動小数点表現により得られた整数関数は、元の初等関数とは全く異なる形になる。図 5 に示すように、関数値の指数部と仮数部が複雑なため、元の初等関数が単調増加であったとしても、得られる整数関数の関数値は、振動する場合がある。そして、符号ビットを考慮した場合、関数はさらに複雑になり、一般に、浮動小数点表現により得られた整数関数の解析は、非常に困難である。し



(a) 浮動小数点表現による整数関数 $f(X)$.



(b) 単調化した関数 $f(Y)$.

図 6: 定義域 $-\infty < X < \infty$ における $f(X) = \exp(X)$ の単調化.

かし, 前節で定義した転位点を考慮することで, 様々な初等関数を M_p 単調増大関数に変換でき, 解析が容易になることを示す. 以下では, M_p 単調増大関数への変換について述べる.

まず, 初等関数の定義域が負の数を含む場合について考える. 浮動小数点表現では負の数は, 符号ビットで表現されるため, 変換された整数関数 $f(X)$ では, 負の数を表現する範囲で X の大小関係が元の数値と逆になり, 元の初等関数の単調性が失われてしまう. そこで, 本稿では,

$$Y = (\overline{x_{n-1}}, x_{n-2} \oplus \overline{x_{n-1}}, x_{n-3} \oplus \overline{x_{n-1}}, \dots, x_0 \oplus \overline{x_{n-1}})_2 \quad (1)$$

を用いた整数関数 $f(Y)$ に変換し, 単調化する. 関数値 (値域) が負の数を含む場合も同様に, 整数関数 g :

$$g = (\overline{f_{n-1}}, f_{n-2} \oplus \overline{f_{n-1}}, f_{n-3} \oplus \overline{f_{n-1}}, \dots, f_0 \oplus \overline{f_{n-1}})_2 \quad (2)$$

を用いて, 単調化できる.

例 3 図 6(a) は, 初等関数 $\exp(X)$ を半精度の浮動小数点表現により整数関数 $f(X)$ に変換したものである. 定義域が負の数を含むため, 元の関数 $\exp(X)$ の単調性が失われている. 図 6(b) は, $f(X)$ を単調化した関数 $f(Y)$ を示している. (例終り)

次に, 考慮すべきことは, p の値の決定法である. 一般に, p の値が大きければ, 転位点の数は小さくなり, 逆に p の値が小さければ転位点の数は大きくなる. 定理 1 は, 任意の p に対して成立するので, より tight な上界を得るために, 本稿では, 整数関数毎に上界が最小となる p を選択する.

表 3 は, 様々な初等関数における転位点の数 k , 変換可能な M_p 単調増大関数, およびその EVBDD 表現に必

表 3: 半精度の浮動小数点関数のクラスと転位点の数 k .

関数 $f(X)$	p	k	定理 1	M_p の上界
$5X + 13.7$	2	1,306	14,324	10,406
$\sin^{-1}(X)$	1	507	5,752	4,231
$\cos^{-1}(X)$	1	616	10,175	8,327
$\tan^{-1}(X)$	1	83	4,480	4,231
$\sinh(X)$	8	368	9,664	8,928
$\cosh(X)$	8	368	9,664	8,928
$\tanh(X)$	1	84	4,483	4,231
$\sinh^{-1}(X)$	1	111	4,564	4,231
$\cosh^{-1}(X)$	1	546	5,869	4,231
$\tanh^{-1}(X)$	2	240	7,030	6,310
2^X	10	132	17,988	17,724
$\exp(X)$	8	483	18,086	17,120
$\ln(X)$	7	397	9,504	8,710
$\log_2(X)$	7	379	9,468	8,710
$1/X$	2	141	6,733	6,310
\sqrt{X}	1	514	5,773	4,231
$1/\sqrt{X}$	1	505	5,746	4,231
$\sqrt{-\ln(X)}$	1	614	6,073	4,231
$X \ln(X)$	5	1,523	11,458	8,412

要な節点数の上界を示している. また, 比較のために, 表 3 の右端に, 転位点を含まない M_p 単調増大関数に必要な節点数の上界も示している. これは, 文献 [18] で, 固定小数点表現の関数を解析するために導出された上界である. 表中で, 単調減少関数は, 系 1 により単調増大関数に変換されている. また, 定義域が負の数を含まない関数や, $f(-X) = -f(X)$ が成立する奇関数は, 表現に符号ビットを含んでいない ($n = 15$ である) ことに注意.

表 4: 半精度の浮動小数点関数を表現する決定グラフの節点数.

関数 $f(X)$	節点数			R_1	R_2
	MTBDD	BMD	EVBDD		
$5X + 13.7$	49,669	12,470	726	1	6
$\sin^{-1}(X)$	30,652	3,618	397	1	11
$\cos^{-1}(X)$	7,198	10,663	861	12	8
$\tan^{-1}(X)$	33,245	10,039	926	3	9
$\sinh(X)$	37,505	10,626	1,114	3	10
$\cosh(X)$	8,715	10,831	1,129	13	10
$\tanh(X)$	31,331	6,805	638	2	9
$\sinh^{-1}(X)$	41,966	10,420	1,533	4	15
$\cosh^{-1}(X)$	12,245	10,363	1,406	11	14
$\tanh^{-1}(X)$	30,613	4,521	541	2	12
2^X	23,112	19,881	1,723	7	9
$\exp(X)$	23,035	23,121	2,356	10	10
$\ln(X)$	27,838	26,173	2,500	9	10
$\log_2(X)$	28,192	16,483	1,163	4	7
$1/X$	52,750	4,259	567	1	13
\sqrt{X}	40,145	5,619	518	1	9
$1/\sqrt{X}$	40,029	6,508	567	1	9
$\sqrt{-\ln(X)}$	9,839	21,837	1,249	13	6
$X \ln(X)$	47,174	22,890	2,833	6	12
平均	30,276	12,480	1,197	6	10

$$R_1 = (EVBDD)/(MTBDD) \times 100$$

$$R_2 = (EVBDD)/(BMD) \times 100$$

決定グラフの変数順序は, 根から $x_{n-1}, x_{n-2}, \dots, x_0$.

表 3 から, 多くの初等関数が, 転位点の数 k と p が小さい M_p 単調増大関数に変換でき, 転位点を含まない M_p 単調増大関数と同程度の大きさの EVBDD でコンパクトに表現できることがわかる. 興味深いことに, 元の初等関数 $X \ln(X)$ は, 単調関数ではないにもかかわらず, 転位点を考慮することで, 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.

表 4 は, 半精度の浮動小数点初等関数を表現する決定グラフの節点数を比較している. 表から, EVBDD の節点数は, 他の表現法に比べ, 1 ~ 2 桁小さいことがわかる.

文献 [18] で示したように, EVBDD を多値決定グラフに拡張する事で, 決定グラフのメモリ量をさらに削減でき, よりコンパクトな表現が得られる. 次の節で, EVMDD に基づく浮動小数点回路の構成と設計法について述べる.

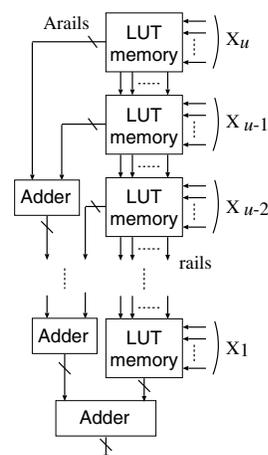


図 7: EVMDD に基づく浮動小数点回路の構成.

4 初等関数の浮動小数点回路

本節では, EVMDD に基づく浮動小数点回路の構成と設計法を示す.

4.1 浮動小数点回路の構成

EVMDD では, 根節点から終端節点までグラフを辿り, 辿った枝の重みの総和を計算することで, 表現している関数の値を計算できるため, 図 7 のようにメモリと整数加算器だけの回路構成で関数値を計算できる. 図 7 の回路構成で, EVMDD は, 各多値変数 X_i 毎の LUT に分割され, 格納される. そのため, 各 LUT への入力は, 多値変数 X_i であり, 隣接する LUT 間の信号線 (rail) は, EVMDD を辿るときの部分関数を表現している. そして, LUT から加算器へ出力される信号線 (Arail) は, 枝の重みを表現している. つまり, 各 LUT は, 次の辿り先アドレスと枝の重みを記憶している.

例 4 図 2(d) の EVMDD を図 7 の回路で実現すると, 各 LUT の内容は, 図 8 のようになる. この図で, r_0 は, EVMDD の部分関数を表現し, a_0 と a_1 は枝の重みを表現している. この EVMDD には, X_2 に関する部分関数が二つしかないため, r_0 は 0 か 1 の値をとる. $r_0 = 0$ のとき表現されている部分関数は, 終端節点なので, 枝の重み a_1 は X_1 の値に依存せず, 常に 0 である. 一方, $r_0 = 0$ のときは, X_1 の値で枝の重み a_1 が決まる. (例終り)

本回路構成の特長を以下に示す.

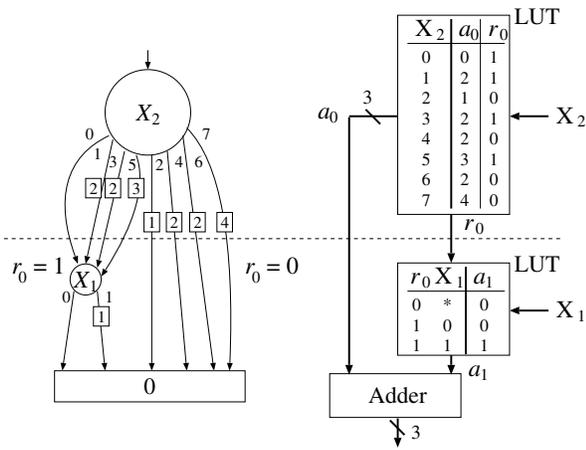


図 8: EVMDD に基づく浮動小数点回路の例.

1. 本回路は、整数関数を表現する EVMDD を辿りながら枝の重みの和を計算するだけなので、整数加算器だけで、浮動小数点関数の値を計算でき、通常の浮動小数点演算に必要な丸め回路や正規化回路などを必要としない。
2. 各 LUT の内容を変更するだけで、多様な初等関数を同じ回路構成で実現できる。
3. EVMDD を用いて浮動小数点表現の初等関数表をそのまま実現しているため、関数近似を用いた回路 [5, 6, 9, 25, 30] に比べ、誤差が小さい。
4. パイプライン処理に適した回路構成のため、高スループットを達成できる。

4.2 EVMDD を用いた浮動小数点回路の設計法

与えられた関数とその定義域、および精度 n に対して、図 7 の回路を自動合成可能である。与えられた初等関数を n ビット精度の浮動小数点表現により整数関数に変換し、その整数関数を表現する EVMDD から図 7 の回路の HDL コードを生成する。生成された回路のメモリ量や性能は、EVMDD のサイズやパス長に大きく依存するため、文献 [17] で提案された MDD のメモリ量最小化アルゴリズムやパス長最小化アルゴリズムが、本浮動小数点回路の最適化にも利用可能である。

3.2 節で述べたように、与えられた初等関数の定義域や値域が負の数を含むとき、図 7 の回路で実現している整数関数は、(1) や (2) によって $f(Y)$ や $g(X)$ に変換さ

表 5: 既存手法との性能比較

浮動小数点回路	スループット [MHz]	遅延時間 [nsec.]
専用回路 [5]	100	39
EVMD	186	27

れている。その場合、本自動合成システムは、元の関数 $f(X)$ を得るために、図 7 の回路の入力に、符号の変換 (2) を実現する回路を加え、出力には、(1) を加える。また、 $f(-X) = -f(X)$ が成立する奇関数では、 $f_s = x_s$ が実現される。ここで、 f_s, x_s は、関数値の符号ビットと X の符号ビットをそれぞれ表す。さらに、 \sqrt{X} のように定義域が負の数を含まない関数では、 $x_s = 1$ のとき、NaN を出力する回路を自動生成する。

4.3 FPGA 実装結果

EVMDD を用いた浮動小数点回路の性能を調べるために、文献 [5] で提案された $\log(X)$ の専用回路と性能比較を行った。半精度の $\log(X)$ を Xilinx Virtex-II FPGA (XC2V1000-4) に実装し、回路のスループットと遅延時間を比較した。比較結果を、表 5 に示す。本提案回路は、多様な初等関数を実現できる汎用回路であるにもかかわらず、 $\log(X)$ の専用回路よりも高いスループットと短い遅延時間を達成した。

以上の結果から、EVMDD は、初等関数の高速かつコンパクトな浮動小数点回路の設計に有用であるといえる。

5 結論とコメント

本稿では、転位点を含む Mp 単調増大関数という新しい関数のクラスを定義し、その関数を表現する EVBDD の節点数の上界を示した。そして、実験により、多くの単調初等関数は、転位点の数と p が小さい Mp 単調増大関数に変換でき、他の表現法に比べコンパクトであることを示した。また、EVMDD に基づく浮動小数点関数回路の設計法を提案し、EVMDD を用いることで従来手法よりも高性能な回路を自動生成できることを示した。

今後は、三角関数などの周期関数のグラフ表現法や浮動小数点回路の設計法について研究を行う予定である。

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 20700051 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] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Trans. Comput.*, Vol. C-35, No. 8, pp. 677–691, Aug. 1986.
- [2] R. E. Bryant and Y.-A. Chen, "Verification of arithmetic circuits with binary moment diagrams," *Design Automation Conference*, pp. 535–541, 1995.
- [3] E. M. Clarke, K. L. McMillan, X. Zhao, M. Fujita, and J. Yang, "Spectral transforms for large Boolean functions with applications to technology mapping," *Proc. of 30th ACM/IEEE Design Automation Conference*, pp. 54–60, June 1993.
- [4] J. Detrey and F. de Dinechin, "Table-based polynomials for fast hardware function evaluation," *16th IEEE Inter. Conf. on Application-Specific Systems, Architectures, and Processors (ASAP'05)*, pp. 328–333, 2005.
- [5] J. Detrey and F. de Dinechin, "A parameterizable floating-point logarithm operator for FPGAs," *Conf. on Signals, Systems and Computers*, pp. 1186–1190, 2005.
- [6] J. Detrey and F. de Dinechin, "A parameterized floating-point exponential function for FPGAs," *IEEE Inter. Conf. on Field-Programmable Technology (ICFPT'05)*, pp. 27–34, 2005.
- [7] R. Drechsler and B. Becker, *Binary Decision Diagrams: Theory and Implementation*, Kluwer Academic Publishers, 1998.
- [8] ANSI/IEEE Standard 754-2008, *IEEE Standard for Floating-Point Arithmetic*, 2008.
- [9] V. K. Jain and L. Lin, "High-speed double precision computation of nonlinear functions," *Proc. of the 12th IEEE Symp. on Computer Arithmetic (ARITH'95)*, Bath, England, pp. 107–114, July 1995.
- [10] 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.
- [11] Y.-T. Lai and S. Sastry, "Edge-valued binary decision diagrams for multi-level hierarchical verification," *Proc. of 29th ACM/IEEE Design Automation Conference*, pp. 608–613, 1992.
- [12] Y.-T. Lai, M. Pedram, and S. B. Vrudhula, "EVBDD-based algorithms for linear integer programming, spectral transformation and functional decomposition," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, Vol. 13, No. 8, pp. 959–975, Aug. 1994.
- [13] D.-U. Lee, W. Luk, J. Villasenor, and P. Y. K. Cheung, "Hierarchical segmentation schemes for function evaluation," *Proc. of the IEEE Conf. on Field-Programmable Technology*, Tokyo, Japan, pp. 92–99, Dec. 2003.
- [14] C. Meinel and T. Theobald, *Algorithms and Data Structures in VLSI Design: OBDD – Foundations and Applications*, Springer, 1998.
- [15] D. M. Miller and M. A. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," *36th International Symposium on Multiple-Valued Logic*, Singapore, May 17–20, 2006.
- [16] J.-M. Muller, *Elementary Function: Algorithms and Implementation*, Birkhauser Boston, Inc., Secaucus, NJ, 1997.
- [17] S. Nagayama and T. Sasao, "On the optimization of heterogeneous MDDs," *IEEE Trans. on CAD*, Vol. 24, No. 11, pp. 1645–1659, Nov. 2005.
- [18] S. Nagayama and T. Sasao, "Representations of elementary functions using edge-valued MDDs," *37th International Symposium on Multiple-Valued Logic*, Oslo, Norway, May 13–16, 2007.
- [19] S. Nagayama and T. Sasao, "Complexities of graph-based representations for elementary functions," *IEEE Trans. on Computers*, (accepted).
- [20] 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 Comp.*, Vol. 54, No. 3, pp. 304–318, Mar. 2005.
- [21] T. Sasao and M. Fujita (eds.), *Representations of Discrete Functions*, Kluwer Academic Publishers 1996.
- [22] T. Sasao, *Switching Theory for Logic Synthesis*, Kluwer Academic Publishers 1999.
- [23] T. Sasao and S. Nagayama "Representations of elementary functions using binary moment diagrams," *36th International Symposium on Multiple-Valued Logic*, Singapore, May 17–20, 2006.
- [24] T. Sasao, S. Nagayama, and J. T. Butler, "Numerical function generators using LUT cascades," *IEEE Transactions on Computers*, Vol. 56, No. 6, pp. 826–838, Jun. 2007.
- [25] M. J. Schulte and E. E. Swartzlander, Jr., "Hardware designs for exactly rounded elementary functions," *IEEE Trans. on Comp.*, Vol. 43, No. 8, pp. 964–973, Aug. 1994.
- [26] M. J. Schulte and J. E. Stine, "Approximating elementary functions with symmetric bipartite tables," *IEEE Trans. on Comp.*, Vol. 48, No. 8, pp. 842–847, Aug. 1999.
- [27] 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, May 2007.
- [28] M. A. Thornton, R. Drechsler, and D. M. Miller, *Spectral Techniques in VLSI CAD*, Springer, 2001.
- [29] I. Wegener, *Branching Programs and Binary Decision Diagrams: Theory and Applications*, SIAM, 2000.
- [30] W. Wong and E. Goto, "Fast evaluation of the elementary functions in single precision," *IEEE Trans. on Comp.*, Vol. 44, No. 3, pp. 453–457, Mar. 1995.
- [31] S. N. Yanushkevich, D. M. Miller, V. P. Shmerko, and R. S. Stankovic, *Decision Diagram Techniques for Micro- and Nanoelectronic Design*, CRC Press, Taylor & Francis Group, 2006.