2003－ARC－153 (4)
2003／5／8

倍精度浮動小数点指数関数計算回路の設計

河瀬 朋範† 高木 直史†† 高木 一義††

概要 指数関数計算は、科学技術計算の分野でしばしば現れる計算である。本報告では、テーブル参照と多項式近似に基づく、指数関数計算回路の設計について検討する。入出力は IEEE754 標準の倍精度浮動小数点基本フォーマットとし、誤差は 1 ulp (unit in the last place) とする。本報告では、近似多項式の次数とテーブルの分割数を変えて、テーブルサイズとクロック · サイクル数を評価する。また、この回路に適した積和演算器の構成を提案する。積和演算器をパイプライン化しない場合と、2 段パイプライン化する場合に対して、回路構成を示す。

Design of Double Precision Floating Point Exponential Function Computing Circuit

TOMONORI KAWASE,† NAOFUMI TAKAGI†† and KAZUYOSHI TAKAGI††

Abstract Exponential function often appears in the field of scientific computing. In this report, we discuss the design of exponential function computing circuit based on table look-up and polynomial approximation. The inputs and the outputs of the circuit are in IEEE-754 double-precision floating-point format. The final error is within 1 ulp (unit in the last place). We evaluated the total table size and clock cycles for several combinations of the degree of approximation polynomial and the number of table division. We propose the structure of multiplier-accumlator suitable for this circuit and show the circuit design with either unfolded or pipelined multiplier-accumulator.

1. はじめに

近年の集積回路技術の発展により、高速な並列乗算器や除算器、開平器などがプロセッサや ASIC に搭載されるようになってきている。今後、さらに LSI の集積度が上がれば、より複雑な演算を行う回路も搭載されるようになると考えられる [1]。そのような演算の一つとして指数関数計算が挙げられる。指数関数計算は、科学技術計算の分野でしばしば現れる計算である [2]。

指数関数計算のアルゴリズムとしては、STL (Seqential Table Look-up) 法 [3]、table-driven アルゴリズム [4][5] 等が提案されている。STL 法はテーブルを逐次参照しながら指数関数計算を行うアルゴリズムである。table-driven アルゴリズムはテーブル参照と多項式近似に基づいて指数関数計算を行うアルゴリズムである。

本報告では、テーブル参照と多項式近似に基づく、指数関数計算回路の設計について検討する。入出力は IEEE754 標準の倍精度浮動小数点基本フォーマットとし、誤差を 1 ulp (unit in the last place) 以内とする。本報告では、近似多項式

† 名古屋大学大学院工学研究科情報工学専攻
Department of Information Engineering, Nagoya University
†† 名古屋大学大学院情報科学研究科情報システム学専攻
Department of Information Engineering, Nagoya University

表 1　IEEE754 倍精度浮動小数点基本フォーマット

$E=2047, F\neq 0$	$x=NaN$	非数
$E=2047, F=0$	$x=(-1)^S\cdot\infty$	$\pm\infty$
$0<E<2047$	$x=(-1)^S\cdot 2^{E-1023}\cdot(1+F)$	正規化数
$E=0, F\neq 0$	$x=(-1)^S\cdot 2^{-1022}\cdot(0+F)$	DEN 数
$E=0, F=0$	$x=(-1)^S\cdot 0$	± 0

表 2　入力 x に対する $\exp(x)$

x	$exp(x)$
NaN	NaN
$+\infty$	$+\infty$
$-\infty$	$+0$
正規化数 ($E>1032$)	$+\infty$
正規化数 ($969\leq E\leq 1032$)	$exp(x)$
正規化数 ($E<969$)	1
DEN 数	1
± 0	1

の次数とテーブルの分割数を変えて、テーブルサイズとクロック · サイクル数を評価する。また、この回路に適した積和演算器の構成を提案する。積和演算器をパイプライン化しない場合と、2 段パイプライン化する場合に対して、回路構成を示す。

2. 倍精度浮動小数点指数関数計算

本報告では、入力 x、出力 $\exp(x)$ はともに、IEEE754 標準の倍精度浮動小数点基本フォーマットで表されているものとする。入力 x を IEEE754 標準の倍精度浮動小数点基本フォーマットで表すと、$x=(-1)^S\cdot 2^{E-1023}\cdot(1+F)$ (S:符号, 1 ビット、E:指数部, 11 ビット、F: 仮数部, 52 ビット) となる。E、F の値により、入力 x は表 1 のように分類される。

まず、入力 x に対する $\exp(x)$ の値について述べる。IEEE754 標準の倍精度浮動小数点基本フォーマットの表現可能最大値は $(2-2^{-52})\times 2^{1023}$ $\left(\approx 1.79763\times 10^{308}\right)$ である。従って、入力 x が $x\geq 710$ の場合、$\exp(x)$ の値がこの表現可能最大値よりも大きくなるので $\exp(x)$ の値は $+\infty$ になる。また、Denomarlized 数 (DEN 数) の表現可能最小値は 2^{-1074} $\left(\approx 4.94065\times 10^{-324}\right)$ である。従って、入力 x が $x\leq -745$ 場合、$\exp(x)$ の値がこの表現可能最小値よりも小さくなるので $\exp(x)$ の値は 0 になる。すなわち、$\exp(x)$ の値が IEEE754 標準の倍精度浮動小数点基本フォーマットで表現可能とするような入力 x の範囲は $-745<x<710$ となり、$|x|<745$ であるから、$E\leq 1032$ となる。

$x\simeq 0$ の場合は、$\exp(x)\simeq 1$ となる。IEEE754 標準で定められている丸めモードのうちの *round to nearest* を適用すると、$1-2^{-54}\leq\exp(x)\leq 1+2^{-53}$ となる入力 x に対しては $\exp(x)=1$ を出力する。そのような入力 x の範囲は $-2^{-54}\leq x\leq 2^{-54}$ である。従って、$E\geq 969$ のとき、$\exp(x)\neq 1$ となる。

以上より、この回路で扱う E の範囲は、$969\leq E\leq 1032$ となる。表 2 に入力 x に対する $\exp(x)$ の値を示す。

次に、$969\leq E\leq 1032$ の場合について、指数関数計算の手順を示す。まず、$\exp(x)$ を以下のように変形する。

$$\begin{aligned}\exp(x)&=\exp\left((-1)^S\cdot 2^{E-1023}\cdot(1+F)\right)\\&=2^{(-1)^S\cdot\frac{2^{E-1023}}{\ln 2}\cdot(1+F)}\end{aligned}\tag{1}$$

ここで、$D=(-1)^S\cdot\frac{2^{E-1023}}{\ln 2}\cdot(1+F)$ とし、D の整数部を D_i、小数部を D_f とすると、

$$\begin{aligned}D&=D_i+D_f\\\exp(x)&=2^D=2^{D_i+D_f}=2^{D_i}\cdot 2^{D_f}\end{aligned}\tag{2}$$

と表せる。

従って、$D_i>-1023$ の場合、$\exp(x)$ の符号ビットは 0、指数部は D_i+1023、仮数部は 2^{D_f} となる。また、$D_i\leq -1023$ の場合、$\exp(x)$ の値は DEN 数となる。このとき、$\exp(x)$ の符号ビットは 0、指数部は 0、仮数部は 2^{D_f} を $(-1022-D_i)$ ビット右シフトした値となる。

3. 計算法と回路構成

3.1 計　算　法

$D=(-1)^S\cdot\frac{2^{E-1023}}{\ln 2}\cdot(1+F)$ を以下の手順で計算して整数部 D_i と小数部 D_f に分離する。

1. $(1+F)$ と $\frac{1}{\ln 2}$ との乗算。
2. 上記の乗算結果を $(E-1023)$ ビットだけシフト (左シフトは最大 9 ビット、右シフトは最大 54 ビット)。
3. 上記のシフトした結果を整数部 D_i と小数部 D_f に分離。($D_i=12$ ビット)
   符号ビット $S=1$ のときは、2 の補数をとる。ここでは、最終桁の誤差は許容範囲であるので、実際には各ビットを反転し、1 の補数をとるだけでよい。

以上のようにして D を分離した結果、式 (2) が得られる。

$\exp(x)$ の仮数部に該当する 2^{D_f} は、小数部 D_f の上位ビットを D_{f_u}、下位ビットを D_{f_l} とすると、

$$2^{D_f}=2^{D_{f_u}+D_{f_l}}=2^{D_{f_u}}\cdot 2^{D_{f_l}} \qquad (3)$$

と表せる。式 (3) の $2^{D_{f_l}}$ は多項式近似して積和演算で求め、$2^{D_{f_u}}$ はテーブル参照で求める。

また、D_{f_u} の上位ビットを $D_{f_{u1}}$、下位ビットを $D_{f_{u2}}$ とすると、テーブル参照で求める $2^{D_{f_u}}$ は、以下のように分割することができる。

$$2^{D_{f_u}}=2^{D_{f_{u1}}}+2^{D_{f_{u2}}}=2^{D_{f_{u1}}}\cdot 2^{D_{f_{u2}}} \qquad (4)$$

従って、分割する前の $2^{D_{f_u}}$ のテーブルの値は、$2^{D_{f_{u1}}}$ のテーブルの値と $2^{D_{f_{u2}}}$ のテーブルの値を乗算して求めることができる。

図 1 に指数関数計算の流れ図を示す。

3.2 回 路 構 成

式 (3) の $2^{D_{f_l}}$ は多項式近似して積和演算で求めるので、積和演算器が必要となる。前節で示した計算法では、複数の乗算を同時に行うことはない。よって、全ての乗算は 1 つの積和演算器で共有する。また、積和演算器を 2 段パイプライン化することにより、テーブル分割で生じる乗算と近似多項式中での乗算がパイプライン処理可能となる。

任意ビットのシフト操作にはバレルシフタが必要となる。シフト操作を同時に行うことはないので、シフト操作は 1 つのバレルシフタで共有する。

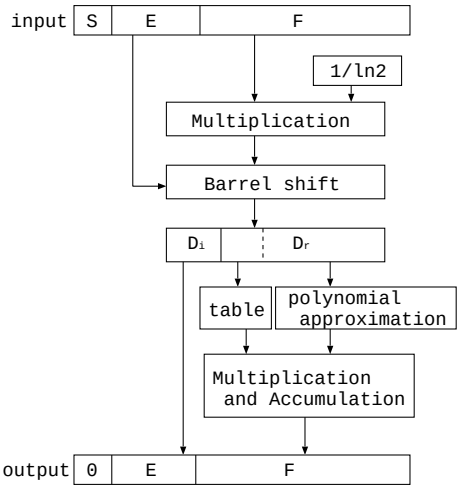

図 1　指数関数計算の流れ図

4. 仮数部の計算

本章では、多項式近似にテーラー展開を用いて、仮数部の計算 (3) における近似多項式の次数とテーブルの分割数について考察する。

ここで、前章で示した小数部 D_f の上位ビット D_{f_u} のビット長を m ビットとし、下位ビット D_{f_l} に対する値 $2^{D_{f_l}}$ をテーラー展開により k 次式で近似する。このとき、$D_{f_l}=y$ とすると最大誤差 ε_{taylor} は、

$$\varepsilon_{taylor}=2^{2^{-m}}-\sum_{i=0}^{k}\frac{1}{k!}\cdot(\ln 2)^i\cdot y^i \qquad (5)$$

となる。この ε_{taylor} が 2^{-52} 以下に収まるように、k に対して m を求めると表 3 に示すようになる。

また、表 3 にはテーブルの分割数に対する、テーブルサイズ (1 エントリを 56 ビットとする) と、仮数部の計算に必要なクロック · サイクル数も示した。表 3 の N_1 は積和演算器をパイプライン化しない場合のクロック · サイクル数、N_2 は積和演算器を 2 段パイプライン化した場合のクロック · サイクル数である。

表 3 から、テーブルの分割数を増やすと、テーブルサイズは削減できるが、仮数部の計算にかかるクロック · サイクル数が増えることがわか

表 3　仮数部計算に必要なサイクル数とテーブルサイズ

次数 (k)	分割数 (m)	D_{f_u}	テーブルサイズ	N_1	N_2
2	1	16	$2^{16}\times 56$	3	6
	2	8+8	$2^8\times 56+2^8\times 48$	4	6
	3	5+5+6	$2^5\times 56+2^5\times 51+2^6\times 46$	5	7
3	1	12	$2^{12}\times 56$	4	8
	2	6+6	$2^6\times 56+2^6\times 50$	5	8
4	1	9	$2^9\times 56$	5	10
	2	4+5	$2^4\times 56+2^5\times 52$	6	10

る。従って、テーブルサイズとクロック · サイクル数を考慮して、最適な近似多項式の次数とテーブルの分割数を決める必要がある。表 3 から考察すると、積和演算器をパイプライン化しない場合は、2 次近似でテーブルを 3 分割にするか、または 3 次近似でテーブルを 2 分割にすればよいと考えられる。一方、積和演算器を 2 段パイプライン化する場合は、2 次近似でテーブルを 3 分割にすればよいと考えられる。

5. 積和演算器

積和演算 $M_1\times M_2+M_3$ は加数 M_3 を部分積として乗算に組み込むことで実現できる。M_1 のビット長を小数点以下 m_1 ビット、M_2 及び M_3 のビット長を小数点以下 m_2 ビットとする。

ここでの M_1 は小数部 D_f の下位ビット D_{f_l} なので、M_1 の上位ビットは 0 である。従って、図 2 に示す部分積マトリックスの下の方は全て 0 となるので、図 2 の斜線部で、M_3 と最下段の部分積との論理和をとることで、積和演算は実現できる。また、$(1+F)$ の乗算と定数 $\frac{1}{\ln 2}$ との乗算は、図 2 に示すように、小数点以下 m_2 ビット目以下で打ち切れば、図 2 の破線部の部分積の生成と累算の必要がない。

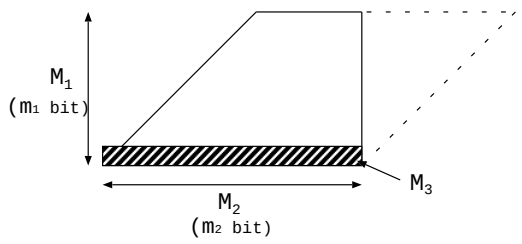

図 2　積和演算器の部分積マトリックス

全ての乗算は 1 つの積和演算器を共有するので、この誤差を考慮して、m_1 及び m_2 を決定する必要がある。

部分積マトリックスの小数点以下 m_2 ビット目以下を打ち切ることによって生じる誤差を E_{trunc} とすれば、

$$E_{trunc}=2^{-m_1-m_2}\cdot\sum_{i=0}^{m_1}(i+1)\cdot 2^i \qquad (6)$$

と表せる。さらに、出力のビット長を m_1+k ビットとし、それ以降を切り捨てることによって生じる誤差を E_{round} とすれば、

$$\begin{aligned}E_{round}&=2^{-m_1-k}\cdot\sum_{i=0}^{m_2-m_1-k-1}2^{-i-1}\\&=2^{-m_1-k}-2^{-m_2}\end{aligned} \qquad (7)$$

と表せる。従って、この積和演算器の出力の誤差 E_{total} は、

$$E_{total}=E_{trunc}+E_{round} \qquad (8)$$

と表せる。

6. 回 路 設 計

表 3 より、小数部 D_f の上位 16 ビットを 5 ビット、5 ビット、6 ビットに 3 分割して、テーブル参照で求め、下位ビットを 2 次近似多項式で求める方法で回路設計を行う。

6.1 誤 差 解 析

誤差を 1 ulp 以内とする回路を設計するためには、2 次近似多項式による誤差を ε_{approx}、仮数部を計算するときに生じる誤差を ε_{comput} とすると

表 4　2 次近似多項式の係数

係数	
A_0	1.00000000000000000000000000000000000000000000000000000000000000
A_1	0.10110001011100100001011111110111110010011101000101011000
A_2	0.00111101011111111001000101001111010011110110011110101110

以下の式を満たす必要がある。

$$\varepsilon_{approx} + \varepsilon_{comput} \leq 2^{-53} \tag{9}$$

ここで、積和演算器の乗数のビット長を小数点以下 m_1 ビット、被乗数のビット長を小数点以下 m_2 ビットとした場合の ε_{comput} について考える。まず、$(1+F)$ と $\frac{1}{\ln 2}$ とを乗算するときに生じる誤差 ε_l は、

$$\begin{aligned}\varepsilon_l &= 2^{-m_2} \cdot 53 + 2^{-m_1-9} - 2^{-m_2} \\ &< 2^{-m_2+6} - 2^{-m_2+3} + 2^{-m_1-9} - 2^{-m_2}\end{aligned} \tag{10}$$

と表せる。これが最大 9 ビット左シフトされるので、シフト後の誤差 ε_y は、

$$\begin{aligned}\varepsilon_y &= \varepsilon_l \cdot 2^9 \\ &< 2^{-m_2+15} - 2^{-m_3+12} + 2^{-m_1} - 2^{-m_2+9}\end{aligned} \tag{11}$$

と表せる。次に 2 次近似多項式 $Y = A_0 + y(A_1 + yA_2)$ を計算するとき、$A_1 + yA_2$ の計算で生じる誤差を ε_{p_1}、$A_0 + y(A_1 + yA_2)$ の計算で生じる誤差を ε_{p_2} とすると、

$$\begin{aligned}\varepsilon_{p_1} &= A_2\varepsilon_y + \varepsilon \\ \varepsilon_{p_2} &= y\varepsilon_{p_1} + (A_1 + yA_2)\varepsilon_y + \varepsilon\end{aligned} \tag{12}$$

と表せる。ここで ε は前節の式 (6) で示した E_{trunc} である。

また、3 つテーブルから参照した値を上位のテーブルから各々C_0、C_1、C_2、各々の値の誤差を $\varepsilon_t (\leq 2^{-m_1-1})$ とする。そして、$Y \times C_2$ を計算するときに生じる誤差を ε_{q_1}、$Y \times C_2 \times C_1$ を計算するときに生じる誤差を ε_{q_2}、$Y \times C_2 \times C_1 \times C_0$ を計算するときに生じる誤差を ε_{q_3} とすると、

$$\begin{aligned}\varepsilon_{q_1} &= Y\varepsilon_t + C_2\varepsilon_{p_2} + \varepsilon \\ \varepsilon_{q_2} &= YC_2\varepsilon_t + C_1\varepsilon_{q_1} + \varepsilon \\ \varepsilon_{q_3} &= YC_2C_1\varepsilon_t + C_0\varepsilon_{q_2} + \varepsilon\end{aligned} \tag{13}$$

と表せる。従って、仮数部を計算するときに生じる誤差 ε_{comput} は、

$$\varepsilon_{comput} = \varepsilon_{q_3} \tag{14}$$

となる。

以上より、式 (9) を満たすには、$m_1 = 56$、$m_2 = 72$、$\varepsilon_{approx} \leq 2^{-56}$ とすればよい。

6.2 近似多項式の係数

実際に使う近似多項式の係数を求めるときには、MiniMax 近似法を用いる。MiniMax 近似法とは最大誤差を最小とする近似多項式を求める手法である。本報告では、MiniMax 近似多項式を求める際に、Mathematica を使用した。

前節で示したように、近似多項式の係数は、近似多項式による最大誤差が 2^{-56} 以下となるように決めなければならない。$0 \leq y \leq 2^{-16}$ において、2^y を 2 次 MiniMax 近似多項式 $A_0 + A_1y + A_2y^2$ で近似したときの最大誤差は、2^{-57} 以下となる。このとき、Mathematica で求められる係数のうち、小数点以下 56 ビット目までを使うと、近似多項式による最大誤差が 2^{-56} 以下となる。従って、回路設計に使う 2 次近似多項式 $A_0 + A_1y + A_2y^2$ の係数は表 4 に示す値になる。

6.3 積和演算器をパイプライン化しない場合

積和演算器をパイプライン化しないで回路設計を行った場合、

1. $(1+F) \times \frac{1}{\ln 2}$: 1 サイクル
2. $(E - 1023)$ バレルシフト+2 の補数 : 1 サイクル
3. テーブル参照 + 多項式近似 : 5 サイクル
4. バレルシフト (DEN 数出力) : 1 サイクル

となり指数関数計算は 8 サイクルで終了する。

6.4 積和演算器を 2 段パイプライン化する場合

積和演算器を 2 段パイプライン化する場合、テーブル分割で生じる乗算と近似多項式中での乗算がパイプライン処理可能となる。図 3 に仮数部の計算手順を示す。

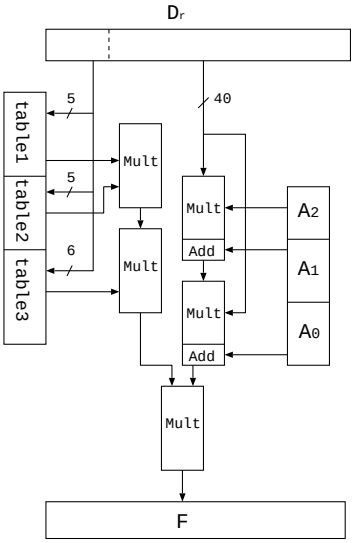

図 3　仮数部の計算

積和演算器を 2 段パイプライン化して回路設計を行った場合、

1. $(1+F)\times\frac{1}{\ln 2}$: 2 サイクル
2. $(E-1023)$ バレルシフト + 2 の補数 : 1 サイクル
3. テーブル参照 + 多項式近似 : 7 サイクル
4. バレルシフト (DEN 数出力) : 1 サイクル

となり、指数関数計算は 11 サイクルで終了する。

積和演算器を 2 段パイプライン化する場合、最終的な指数関数計算回路のブロック図は図 4 のようになる。

7. ま と め

本報告では、入出力は IEEE754 標準の倍精度浮動小数点基本フォーマットとし、テーブル参照と多項式近似に基づく指数関数計算回路の設計について検討した。本報告では、近似多項式の次数とテーブルの分割数を変えて、テーブルサイズとクロック · サイクル数を評価した。また、この回路に適した積和演算器の構成を提案した。積和演算器をパイプライン化しない場合と、2 段パイプライン化する場合に対して、回路構成を示した。本報告で示した手法を用いれば、誤差を 1 ulp (unit in the last place) 以内とする指数関数計算回路が設計できる。

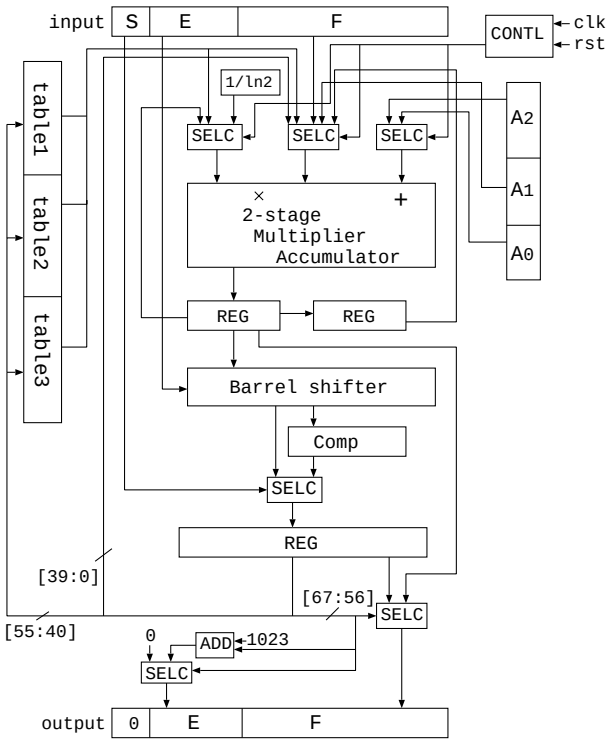

図 4　指数関数計算回路ブロック図

参 考 文 献

[1] 高木直史, ” 初等関数計算回路のアルゴリズム”, 情報処理学会, 第 37 巻 第 4 号, 1996.

[2] John Harrison, ”The Computation of Transcendental Function on IA-64 Architecture”, Intel Technology Journal Q4 , 1999.

[3] Chen, T.C . , ”Automatic Computation of Exponemtials, Logarithms, Rations and Square Roots”,IBM J.Research and Development, Vol.16, No. 4, pp. 380-388 (July 1972).

[4] Douglas M.Priest, ”Fast Table-Driven Algorithms for Interval Elementary Functions” , Proc.13th IEEE Symposium on Computer Arithmetic 1997.

[5] J.M.Muller, ”Elementary Function-Algorithms and Implementations”,Birkhauser,1997.