

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3884809号
(P3884809)

(45) 発行日 平成19年2月21日(2007.2.21)

(24) 登録日 平成18年11月24日(2006.11.24)

(51) Int. Cl.

G06F 7/552 (2006.01)

F I

G06F 7/552

A

請求項の数 2 (全 14 頁)

<p>(21) 出願番号 特願平9-7240 (22) 出願日 平成9年1月20日(1997.1.20) (65) 公開番号 特開平10-207694 (43) 公開日 平成10年8月7日(1998.8.7) 審査請求日 平成13年3月22日(2001.3.22) 審査番号 不服2004-3721(P2004-3721/J1) 審査請求日 平成16年2月26日(2004.2.26)</p>	<p>(73) 特許権者 000005108 株式会社日立製作所 東京都千代田区丸の内一丁目6番6号 (74) 代理人 100100310 弁理士 井上 学 (72) 発明者 安部 雄一 茨城県日立市大みか町七丁目1番1号 株式会社 日立製作所 日立 研究所内 (72) 発明者 藤田 良 茨城県日立市大みか町七丁目1番1号 株式会社 日立製作所 日立 研究所内</p>
--	---

最終頁に続く

(54) 【発明の名称】 デジタルべき乗演算装置及びそれを用いたグラフィックスシステム

(57) 【特許請求の範囲】

【請求項1】

鏡面指数値 N と各々の画素における法線ベクトルと光源ベクトルの内積である入力値 X を用いて、光源計算部において X^N を求める光源計算を行い、この計算結果に基づいて表示すべき図形データを画素情報に展開するレンダリングプロセッサであって、

前記光源計算部は、入力値 X に対する対数値を対数テーブルを用いて出力する対数算出部と、前記対数算出部の出力と前記光源テーブルからの値 N を乗算する乗算器と、前記乗算器の出力に対する指数値を指数テーブルを用いて出力する指数算出部とを有し、

前記対数算出部は、

入力値域を制限された定義域とする前記対数テーブルと、

前記対数算出部への入力値が前記対数テーブルの入力値域に入るよう、当該入力値に 2^L (L は整数) を乗算して、前記対数テーブルに出力する対数シフト部と、

前記対数テーブルの出力に L を加算して前記対数算出部の出力とする対数加算部を有し、

前記指数算出部は、

入力値域を制限された定義域とする前記指数テーブルと、

前記乗算器からの出力値が前記指数テーブルの入力値域に入るよう、当該入力値から M (M は整数) を減算し、前記指数テーブルに出力する指数減算部と、

前記指数テーブルの出力値に 2^{-M} を乗算して前記指数算出部の出力とする指数シフト部を有することを特徴とするグラフィックスシステム。

10

20

【請求項 2】

請求項 1 において、前記対数算出部により算出される対数の底と、前記指数算出部により算出される指数の底は同一の値であることを特徴とするグラフィックスシステム。

【発明の詳細な説明】**【0001】****【発明の属する技術分野】**

本発明は演算中にべき乗が含まれる処理を実行する情報処理装置に関する。

【0002】**【従来の技術】**

従来、2つのデジタル数値データ X 、 N に対して X^N を求めるべき乗計算については、
技術評論社より平成 3 年 2 月 25 日に発行された奥村晴彦著、「C 言語による最新アルゴリズム事典」の p 105 - 106、p 162 - 163、及び p 304 に記されているように、対数関数及び、指数関数をべき級数展開もしくは、連分数展開することで、対数及び、指数をループ計算によって求める手法を用い、ソフトウェア的に X の対数を計算し、その結果に N を乗算し、最後にこの乗算結果の指数を計算するといった方法が取られていた。

10

【0003】

また別の方法としては、前記 2つのデジタル数値データ X 、 N から直接アドレスを生成してべき乗テーブル (ROM, RAM) を参照する方法が取られていた。

【0004】

20

【発明が解決しようとする課題】

上記のように、従来例では、前者の場合、ループ計算が発生し処理の高速化が難しく、後者の場合、テーブルの入力が X 及び N の 2つであるためテーブルの入力値の階調数は X の階調数と N の階調数の積となりテーブルの容量が大きくなるという問題があった。

【0005】

本発明の目的は、ループ計算を用いず容量の小さいテーブルを参照して高速にべき乗計算を行い得るべき乗演算装置とそれを用いたグラフィックスシステムを提供することにある。

【0006】**【課題を解決するための手段】**

30

本発明の特徴は、入力値 X に対する対数値を対数テーブルを用いて出力する対数算出部と、対数算出部の出力とべき乗する値 N を乗算する乗算器と、この乗算器の出力に対する指数値を指数テーブルを用いて出力する指数算出部とからデジタルべき乗演算装置を構成し、前記対数算出部により算出される対数の底と、前記指数算出部により算出される指数の底を同一の値にしたことにある。

【0007】

また、更にテーブルの容量を減らすために本発明では、前記対数算出部に、その入力値が前記対数テーブルの入力値域に含まれない場合に前記対数算出部の入力値に適当な整数 L に対し 2^L を乗算する対数シフト部と、前記乗算結果を前記対数テーブルの入力とし対数テーブル参照後、参照値に L を加算して前記対数算出部の出力とする対数加算部を設けたものである。

40

【0008】

また、前記指数算出部に、その入力値が前記指数テーブルの入力値域に含まれない場合に前記指数算出部の入力から適当な整数 M を減算する指数減算部と、該減算結果を前記指数テーブルの入力とし指数テーブル参照後、該参照値に 2^{-M} を乗算して前記指数算出部の出力とする指数シフト部を設けたものである。

【0009】

但し、ここでいう対数、指数テーブルとは、RAM, ROM だけに留まらず、入力値に対する対数関数、指数関数の値を一定時間内に算出する回路一般を指すものである。

【0010】

50

本発明では、入力値 X , N に対して X^N のべき乗計算を行う際、対数テーブル参照によって a を底とする対数 $\log a X$ を求め、乗算器で $\log a X \times N$ を計算し、指数テーブル参照によって a の $\log a X \times N$ べき $a^{\log a X \times N} = X^N$ を算出する。本発明ではループ計算を行わないため、高速計算が可能である。また、対数テーブルと指数テーブルとの2つにテーブルを分割することで、各テーブルの入力を1つにできて、テーブルの容量を減らすことができる。

【0011】

また、更にテーブルの容量を減らすために、前記対数算出部の入力値が前記対数テーブルの入力値域に含まれない場合に該入力値に適当な整数 L に対し 2^L を乗算し、該乗算結果を前記対数テーブルの入力とし対数テーブル参照後、該参照値に L を加算して前記対数算出部の出力とし、前記指数算出部の入力値が前記指数テーブルの入力値域に含まれない場合に該入力値から適当な整数 M を減算し、

該減算結果を前記指数テーブルの入力とし指数テーブル参照後、該参照値に 2^{-M} を乗算して前記指数算出部の出力とする。これによって、前記対数算出部、指数算出部の入力値が前記対数、指数テーブルの入力値域に含まれない場合もべき乗計算できる。従って、前記対数、指数テーブルの入力値域を制限することができ、テーブルの容量を減らすことができる。

【0012】

【発明の実施の形態】

以下本発明の実施例を図1～図9を用いて説明する。図8に本発明に基づくデジタルべき乗演算装置を用いた一実施例であるグラフィックス・システムの構成を示す。本システムはアプリケーションソフト等を実行するCPU(1000)、主メモリMM(3000)等を制御するメモリ・コントローラMC(2000)、システムバスを制御するシステムバス・コントローラ(4000)、システムバス・コントローラから受け取ったデータを、座標変換等を行うジオメトリプロセッサ(5000)に送り、またジオメトリプロセッサから帰ってきたデータに対しFI変換、パック、光源計算等の処理を施すGPIF(0000)、GPIF(0000)から送られたデータを画素情報展開するレンダリングプロセッサ(6000)、レンダリングプロセッサ(6000)が生成した画素情報を記憶するフレームメモリ(7000)、及びフレームメモリ(7000)の内容を表示するCRT(8000)からなる。

【0013】

次に、システム全体の動作について説明する。CPU(1000)はアプリケーションを実行し、グラフィックス・コマンドと描画する図形の頂点座標、法線ベクトル、テクスチャ・データ、材質の各反射係数、光源の各反射光用の色等のデータを発行し、MC(2000)とシステムバス・コントローラ(4000)を介してGPIF(0000)に出力する。GPIF(0000)はシステムバス・コントローラ(4000)から送られた前記コマンドとデータをGPIF入力手段(100)に保持する。

【0014】

ジオメトリプロセッサ(5000)はGPIF入力手段(100)に保持しているコマンドとデータを読み、前記コマンドとデータに従い座標変換等の幾何計算を行い、頂点座標、法線ベクトル、テクスチャ・データ等を算出してGPIF(0000)に送る。

【0015】

GPIF(0000)はジオメトリプロセッサ(5000)から送られたデータに対し、前記コマンドとデータに従い必要ならばFI変換、パックを施し、頂点毎の色を計算する光源計算を行い、連続三角形描画コマンド、頂点座標、色、テクスチャ・データをレンダリングプロセッサ(6000)に出力する。

【0016】

レンダリングプロセッサ(6000)は前記コマンドとデータから内挿補間により図形の内部の画素を生成し、CRT(8000)に表示する内容をビットマップ形式でフレームメモリ(7000)に書き込み、画像をCRT(8000)に表示する。

10

20

30

40

50

【 0 0 1 7 】

更に、G P I F (0 0 0 0) の内部構成について詳細に説明する。

【 0 0 1 8 】

G P I F (0 0 0 0) は前記システムバス・コントローラ (4 0 0 0) から送られたコマンドとデータを保持するバッファであるG P I F入力手段(1 0 0)と、前記コマンドとデータを読み幾何計算を行うジオメトリプロセッサ(5 0 0 0)から送られたデータを保持するバッファであるL B u f (2 0 0) と、前記コマンドとデータをL B u f (2 0 0) からコマンド解釈手段 (6 0 0) 及びF I変換手段 (4 0 0) に出力するためのレジスタであるB u f S W (3 0 0) と、前記コマンドを解釈するコマンド解釈手段 (6 0 0) と、前記コマンドに従い必要ならデータのF I変換を行うF I変換手段 (4 0 0) と、前記コマンドに従い必要なら前記F I変換後のデータのパック処理を行うパック手段 (5 0 0) と、前記F I変換、パック処理後の光源計算に必要な光源データを保持する光源テーブル (7 0 0) と、光源テーブル (7 0 0) の保持する光源データを基に光源計算を行い色を算出する光源計算手段 (0 0 0) と、ジオメトリプロセッサ (5 0 0 0) , パック手段 (5 0 0) 及び光源計算手段 (0 0 0) から送られたコマンドとデータの順序を制御する制御手段 (8 0 0) と、前記コマンドとデータを保持するバッファであるC B u f (9 0 0) と、前記コマンドとデータをレンダリングプロセッサ (6 0 0 0) に出力するためのレジスタであるB u f F L (9 5 0) から構成される。

10

【 0 0 1 9 】

前記光源テーブル (7 0 0) 及び光源計算手段 (0 0 0) の詳細を図9に示す。

20

【 0 0 2 0 】

光源テーブル (7 0 0) には光源計算に必要なパラメータが固定小数点数で保持されている。このパラメータは光源に非依存なものと光源に依存して値の変化するものとがある。光源テーブル (7 0 0) は光源に非依存なパラメータの値をそれぞれ1個ずつ、光源に依存して値の変化するパラメータの値をそれぞれ8個ずつ (8光源分) 保持している。もし、光源数が9個以上ある場合は既に計算に使われた値から順に新たな光源の値に一つずつ更新される。

【 0 0 2 1 】

光源に非依存なパラメータに対して、このような書き込み制御を行うために、8個の値のうち、現在何番目の値を計算中であることを示すリードポインタ、R P N Tレジスタが用意されており、R P N T以降の値はロックされ、更新が延期される。

30

【 0 0 2 2 】

光源計算手段 (0 0 0) は法線ベクトルとハーフウェイベクトルの内積を計算するH N内積算出部 (0 1 0) と、該内積のS M乗を計算するべき乗算出部 (0 0) と、法線ベクトルと光源ベクトルの内積を計算するL N内積算出部 (0 2 0) と、べき乗算出部 (0 0) とL N内積算出部 (0 2 0) の出力を用いて各頂点毎の色を算出する色算出部 (0 3 0) から構成される。

【 0 0 2 3 】

H N内積算出部 (0 1 0) は法線ベクトル (N_x, N_y, N_z) とハーフウェイベクトル (H_x, H_y, H_z) の内積を計算し結果13bitをべき乗算出部 (0 0) に出力する。

40

【 0 0 2 4 】

べき乗算出部 (0 0) はH N内積算出部 (0 1 0) の出力を材質の鏡面指数S M (1 から128までの整数) 乗して結果8bitを色算出部 (0 3 0) に出力する。

【 0 0 2 5 】

L N内積算出部 (0 2 0) は法線ベクトル (N_x, N_y, N_z) と光源ベクトル (L_x, L_y, L_z) の内積を計算し結果を色算出部 (0 3 0) に出力する。色算出部 (0 3 0) はR G Bの3色をそれぞれ独立に計算するため、同様のリソースを3セット有している。例えばRについては、環境反射光のR成分L c a R , 拡散反射光のR成分L c d R , 鏡面反射光のR成分L c s R , 環境反射係数のR成分K a R , 拡散反射係数のR成分K d R , 鏡面反射係数のR成分K s R , 放射反射光と全体の環境反射光のR成分の和K R , 減衰係数

50

とスポット光源効果の積 $A t S p$, べき乗算部 (0 0) の出力、及び $L N$ 内積算部 (0 2 0) の出力を入力とし、頂点の色の R 成分 8 bit を出力する。

【 0 0 2 6 】

図 1 にべき乗算部 (0 0) の構成を示す。説明の都合上、入力は X 、及び N とし、出力は X^N とする。つまり、 X 、及び N は上述の説明に於ける $H N$ 内積算部 (0 1 0) の出力、及び材質の鏡面指数 $S M$ に対応する。 X は 1 3 bit の固定小数点数で範囲は $0 \sim 1$ 、 N は 8 bit の固定小数点数で範囲は $0 \sim 1 2 8$ 、 X^N は 8 bit の固定小数点数で範囲は $0 \sim 1$ である。

【 0 0 2 7 】

この回路は入力 X に対する対数関数の値を 1 5 bit の固定小数点数で算出する対数算出部 (1 0) , 該対数算出部 (1 0) の出力と N を乗算し、1 0 bit の固定小数点数を出力する乗算器 (2 0) , 該乗算器 (2 0) の出力に対する指数関数の値を 8 bit の固定小数点数で算出する指数算出部 (3 0) からなる。

10

【 0 0 2 8 】

ここで、対数算出部 (1 0) 及び指数算出部 (3 0) をそのままテーブルにしてしまうと、対数テーブルは入力レンジが $0 \sim 1$ で 1 3 bit、出力レンジが $0 \sim 8$ (厳密には 8 は含まない) で 1 5 bit、指数テーブルは入力レンジが $0 \sim 8$ (厳密には 8 は含まない) で 1 0 bit、出力レンジが $0 \sim 1$ で 8 bit となり、メモリ容量換算でそれぞれ、1 2 2 , 8 8 0 bit , 8 , 1 9 2 bit と膨大な容量になってしまう。

【 0 0 2 9 】

20

しかし、対数及び指数のテーブルを縮退、つまり入出力レンジを制限し、対数算出部 (1 0) 及び指数算出部 (3 0) を以下のように構成することにより、それぞれのテーブルの容量の大幅な低減 (メモリ容量換算で 2 4 , 5 7 6 bit , 7 6 8 bit) と、縮退以前と同様の精度での計算が可能となる。

【 0 0 3 0 】

即ち、該対数算出部 (1 0) は入力値を縮退した対数テーブル (1 2) の入力レンジに入るまで K ビット左シフト (2^K を乗算) し、3 bit のシフト量 K と 1 1 bit のシフト結果を出力する対数シフト部 (1 1) と、該シフト結果に対する対数関数の値を 1 2 bit の固定小数点数として出力する縮退した対数テーブル (1 2) と、該対数テーブル (1 2) の出力に K を加算して、1 5 bit の固定小数点数を出力する対数加算部 (1 3) からなる。

30

【 0 0 3 1 】

また、指数算出部 (3 0) は、入力値から縮退した指数テーブル (3 2) の入力レンジに入るまで M を減じ、3 bit の減算量と 7 bit の減算結果を出力する指数減算部 (3 1) と、該減算結果に対する指数関数の値を 6 bit の固定小数点数として出力する縮退した指数テーブル (3 2) と、該指数テーブル (3 2) の出力を M ビット右シフトする指数シフト部 (3 3) から構成される。

【 0 0 3 2 】

図 2 を使って前記対数算出部 (1 0) が入力 $P x$ に対して出力 $P y$ を算出する際 (この操作を白貫矢印で表している) の動作を示す。図 2 のグラフは底を $2^{-1} = 0.5$ とする定義域 $0 \sim 1$ (厳密には 0 は含まない)、値域 $0 \sim 8$ (厳密には 8 は含まない) の対数関数の一部を表わしたものである。ここで、定義域とは入力値 x の変域を意味し、値域とは x の変動に伴う出力値 y の変域を意味する。

40

【 0 0 3 3 】

領域 0 は定義域 $0.5 \sim 1$ (厳密には 0.5 は含まない)、値域 $0 \sim 1$ (厳密には 1 は含まない) の部分である。対数テーブル (1 2) はこの範囲の対数関数を保持している。つまり、グラフ全体の定義域が $0 \sim 1$ であるのに対して、対数テーブル (1 2) が保持している範囲の定義域は $0.5 \sim 1$ と $1/2$ に、またグラフ全体の値域が $0 \sim 8$ であるのに対して、対数テーブル (1 2) が保持している範囲の値域は $0 \sim 1$ と $1/8$ に縮対している。

【 0 0 3 4 】

領域 1 は定義域 $0.25 \sim 0.5$ (厳密には 0.25 は含まない)、値域 $1 \sim 2$ (厳密には

50

2は含まない)の部分であり、対数関数の性質より領域1は領域0に対し x を 2^{-1} 倍し、 y に1を加えたものである。一般的に領域 K (K は0から7までの整数)は定義域 $2^{-K-1} \sim 2^{-K}$ (厳密には 2^{-K-1} は含まない)、値域 $K \sim K+1$ (厳密には $K+1$ は含まない)の部分であり、対数関数の性質より領域 K は領域0に対し x を 2^{-K} 倍し、 y に K を加えたものである。対数シフト部(11)は $P \times$ がどの領域 K の定義域に含まれているかによって、 $P \times$ を 2^K 倍(K 左シフト)し、領域0の定義域までシフトする。簡単のため $P \times$ は領域1の定義域に含まれているものとし、 $P \times$ を 2^1 倍(1シフト)した結果を $Q \times$ とする(この操作を矢印(1)で表している)。 $Q \times$ は対数テーブル(12)の入力値域に含まれているので、対数テーブル(12)を参照して $Q y$ を得る(この操作を矢印(2)で表している)。最後に対数加算部(13)は $Q y$ にシフト量の1を足して $P y$ を算出する(この操作を矢印(3)で表している)。

10

【0035】

図3を使って対数シフト部(11)の動作を示す。対数シフト部(11)は領域 K の定義域内の入力値が領域0の定義域に入るまで左シフトさせ、そのときのシフト量とシフト結果を出力する。

【0036】

例えば、領域2の定義域は $2^{-3} \sim 2^{-2}$ で13bitの固定小数点数で表わすと0.001000000001~0.010000000000であるが、この定義域にある値0.001010011101を領域0の定義域0.100000000001~1.000000000000まで左シフトさせる場合のシフト量は、この値0.001010011101から0.000000000001を引いた0.001010011100の最上位にある1が上位から2桁目に来るまで左シフトさせた際のシフト量と一致する。この場合シフト量は2である。ここで、0.000000000001を引くのは0.010000000000のように領域内の最大値も例外なく扱うためである。このような場合は0.000000000001を引かずに最上位にある1が上位から2桁目に来るまでシフトさせると0.100000000000となり領域0の定義域に含まれなくなる。

20

【0037】

また、領域0の定義域は0.5~1(厳密には0.5は含まない)13bitであるが、0.5即ち13bitの固定小数点数0.100000000001を引いて定義域を0~0.5(厳密には0.5は含まない)としておくことで上位2bitは必ず00となる。このことを利用して、前記対数テーブル(12)の入力を13bitから、必ず00となる上位2bitを取り去り下位の11bitとすることで、入力ビット数を2bit節約できる。従って、シフト結果から13bitの固定小数点数0.100000000001を引いて上位2bitを取り去った11bitの値を対数テーブル(12)への出力とする。

30

【0038】

但し、シフト量は最大でも7とする。その理由は7bit左シフトしても領域0の定義域に含まれない値は 2^{-8} より小さく、8bit精度のべき乗結果には現われてこないためである。このような場合、13bitの固定小数点数0.100000000001を引くと0未満になるため、0クランプして出力値は0.000000000000とする。

【0039】

(a)の場合、入力値は0.001001110100で0.000000000001を引いた値は0.001001110011である。この値の最上位にある1は2bit左シフトすれば上位から2桁目に来るので、シフト量は2である。従って、入力値0.001001110100を2bit左シフトした0.100111010000がシフト結果となる。出力値はシフト結果0.100111010000から0.100000000001を引いた0.000111001111である。

40

【0040】

(b)の場合、入力値は0.000000100000で0.000000000001を引いた値は0.000000011111である。この値の最上位にある1は7ビット左シフトすれば上位から2桁目に来るので、シフト量は7である。従って、入力値0.000000100000を7ビット左シフトした1.000000000000がシフト結果となる。出力値はシフト結果1.000000000000から0.100000000001を引いた0.011111111111である。

【0041】

50

(c) の場合、入力値は0.000000000101で0.000000000001を引いた値は 0.000000000100である。この値の最上位にある1は7ビット左シフトしても上位から2桁目に来ることはないので、シフト量は最大の7である。従って、入力値0.000000000101を7ビット左シフトした0.001010000000がシフト結果となる。シフト結果1.000000000000から0.100000000001を引くと0未満となるので0クランプして出力値は0.000000000000となる。

【0042】

上述のように動作する対数シフト部(11)の回路図を図4に示す。

【0043】

対数シフト部(11)は上述のように入力値から13bitの固定小数点数 0.0000000000001を引いた値を用いてシフト量を決定するため入力直後にこの引き算を行う。図4の最上段には該引き算結果の上位8bitと入力値を並べている。シフトに関する論理は大きく3段に分かれている。まず、1段目ではNOR1が該引き算結果の上位8bitのうち、上位5bitのNORをとりこの値の0,1に応じて該引き算結果の上位8bit及び入力値を4bit左シフトするか否かを決定する。

【0044】

もしNOR1の出力が1であれば、該引き算結果の上位5bitが全て0であり、4bit左シフトする余地があることを意味しているため、該引き算結果の上位8bit及び入力値を4bit左シフトする。また、シフト量の最上位を1とする。これは4bit左シフトしたことを示す。

【0045】

もし、NOR1の出力が0であれば該引き算結果の上位5bitの中に1が含まれていて、4bit左シフトはできないことを意味しているため、該引き算結果の上位8bit及び入力値は左シフトしない。また、シフト量の最上位を0とする。これは4bitは左シフトできなかったことを示す。

【0046】

次に、2段目ではNOR2が該引き算結果の1段目におけるシフト結果の上位3bitのNORをとり、この値の0,1に応じて該引き算結果及び入力値の1段目におけるシフト結果を更に2bit左シフトするか否かを決定する。

【0047】

もしNOR2の出力が1であれば、該引き算結果の1段目におけるシフト結果の上位3bitが全て0であり、2bit左シフトする余地があることを意味しているため、該引き算結果及び入力値の1段目におけるシフト結果を2bit左シフトする。また、シフト量の第2桁目を1とする。これは2bit左シフトしたことを示す。

【0048】

もしNOR2の出力が0であれば、該引き算結果の1段目におけるシフト結果の上位3bitの中に1が含まれていて、2bit左シフトはできないことを意味しているため、該引き算結果及び入力値の1段目におけるシフト結果は左シフトしない。また、シフト量の第2桁目を0とする。これは2bit左シフトできなかったことを示す。

【0049】

次に、3段目ではNOR3が該引き算結果の2段目におけるシフト結果の上位2bitのNORをとり、この値の0,1に応じて該引き算結果及び入力値の2段目におけるシフト結果を更に1bit左シフトするか否かを決定する。

【0050】

もしNOR3の出力が1であれば、該引き算結果の2段目におけるシフト結果の上位2bitが全て0であり、1bit左シフトする余地があることを意味しているため、該引き算結果及び入力値の2段目におけるシフト結果を1bit左シフトする。また、シフト量の最下位を1とする。これは1bit左シフトしたことを示す。

【0051】

もしNOR3の出力が0であれば、該引き算結果の2段目におけるシフト結果の上位2bitの中に1が含まれていて、1bit左シフトはできないことを意味しているため、該引き

10

20

30

40

50

算結果及び入力値の2段目におけるシフト結果は左シフトしない。また、シフト量の最下位を0とする。これは1bit左シフトできなかったことを示す。

【0052】

この段階でシフト量3bitは決定されるが、対数テーブルへの出力値は入力値の3段目におけるシフト結果から13bitの固定小数点数0.100000000001を引き算し更に0クランプした値となる。

【0053】

次に対数テーブル(12)について説明する。対数テーブル(12)の入力は上述のように入力値域0~0.5(厳密には0.5は含まない)の11bitの固定小数点数である。また、対数テーブル(12)の出力は入力値に13bitの固定小数点数0.100000000001を足した値に於ける対数関数の値を12bitの固定小数点数で表わしたものであり、出力値域は0~1(厳密には1は含まない)である。

10

【0054】

対数テーブル(12)はRAMやROMで作成し、入力値をアドレスに変換して参照するように構成することもできるが、ここでは、出力論理値を入力論理値の論理式で表現して論理式に対応する回路で対数テーブル(12)を構成する。

【0055】

対数テーブル(12)の入力の各bitを a_0, a_1, \dots, a_{10} とし、対数テーブル(12)の出力の各bitを b_0, b_1, \dots, b_{11} とすると、各 b_0, b_1, \dots, b_{11} は a_0, a_1, \dots, a_{10} の積和の論理式で表わすことができる。更に、この積和の各項を主項とする方法として、クイーンの方法や、コンセンサス法が著名である。クイーンの方法や、コンセンサス法については丸善株式会社が昭和57年6月30日に発行した後藤宗弘著、電気・電子学生のための計算機工学p40~45に示されている。

20

【0056】

このような方法で生成された論理式に対応する回路で対数テーブル(12)を構成することができる。

【0057】

実際に論理合成してみた結果、0.35 μ mのCMOSで約4kゲートを要した。

【0058】

最後に対数加算部(13)について説明する。対数加算部(13)の入力は対数シフト部(11)で算出したシフト量と対数テーブル(12)の出力である。対数加算部(13)は対数テーブル(12)の出力値に該シフト量を加算して出力する。

30

【0059】

テーブルの出力値域は0~1(厳密には1は含まない)であり、シフト量は整数であるから、対数加算部(13)の出力はテーブルの出力値12bitの上位にシフト量の3bitを付け足した15bitの固定小数点数である。

【0060】

次に前記乗算器(20)について説明する。該乗算器(20)の入力は前記対数算出部(10)の出力と、Nである。

【0061】

該乗算器(20)は前記対数算出部(10)の出力15bitとN8bitを乗算して出力値域は0~8(厳密には8は含まない)の10bitの固定小数点数として出力する。

40

【0062】

但し、乗算の結果が8以上になった場合は最大出力値にクランプする。その理由は、 2^{-1} の8以上のべきは 2^{-8} より小さく、8bit精度のべき乗結果には現われてこないためである。

【0063】

図5を使って前記指数算出部(30)が入力Pxに対して出力Pyを算出する際(この操作を白貫矢印で表している)の動作を示す。図5のグラフは底を $2^{-1} = 0.5$ とする定義域0~8(厳密には8は含まない)、値域0~1(厳密には0は含まない)の指数関数の

50

一部を表わしたものである。領域 0 は定義域 $0 \sim 1$ (厳密には 1 は含まない)、値域 $0.5 \sim 1$ (厳密には 0.5 は含まない) の部分であり、指数テーブル (32) はこの範囲の指数関数を保持している。つまり、グラフ全体の定義域が $0 \sim 8$ であるのに対して、指数テーブル (32) が保持している範囲の定義域は $0 \sim 1$ と $1/8$ に、またグラフ全体の値域が $0 \sim 1$ であるのに対して、指数テーブル (32) が保持している範囲の値域は $0.5 \sim 1$ と $1/2$ に縮退している。

【0064】

領域 1 は定義域 $1 \sim 2$ (厳密には 2 は含まない)、値域 $0.25 \sim 0.5$ (厳密には 0.25 は含まない) の部分であり、指数関数の性質より領域 1 は領域 0 に対し x に 1 を加え、 y を 2^{-1} 倍したものである。

10

【0065】

一般的に領域 M (M は 0 から 7 までの整数) は定義域 $M \sim M+1$ (厳密には $M+1$ は含まない)、値域 $2^{-M-1} \sim 2^{-M}$ (厳密には 2^{-M-1} は含まない) の部分であり、指数関数の性質より領域 M は領域 0 に対し x に M を加え、 y を 2^{-M} 倍したものである。

【0066】

指数減算部 (31) は Px がどの領域 M の定義域に含まれているかによって、 Px から M を減算し、領域 0 の定義域までスライドする。簡単のため Px は領域 1 の定義域に含まれているものとし、 Px から 1 減算した結果を Qx とする (この操作を矢印 (1) で表している)。 Qx は指数テーブル (32) の入力値域に含まれているので、指数テーブル (32) を参照して Qy を得る (この操作を矢印 (2) で表している)。最後に指数シフト部 (33) は Qy に減算量の 1 だけ右シフト (2^{-1} を乗算) して Py を算出する (この操作を矢印 (3) で表している)。

20

【0067】

指数減算部の説明をする。指数減算部 (31) の入力を入力値域 $0 \sim 8$ (厳密には 8 は含まない) の 10 bit の固定小数点数である。上述のように、指数減算部 (31) はその入力値がどの領域 M の定義域に含まれているかによって、 Px から M を減算し、領域 0 の定義域までスライドするが、 M は入力値の上位 3 bit であり、入力値から M を引いた値は入力値の下位 7 bit である。

【0068】

次に指数テーブル (32) について説明する。指数テーブル (32) の入力指数減算部 (31) の出力であり、入力値域 $0 \sim 1$ (厳密には 1 は含まない) の 7 bit の固定小数点数である。また、領域 0 の値域は $0.5 \sim 1$ (厳密には 0.5 は含まない) であるが、 y 方向に -0.5 平行移動して値域 $0 \sim 0.5$ (厳密には 0.5 は含まない) とすることで、指数テーブル (32) の出力の上位 2 bit が 00 となり、出力 bit 数を 2 bit 減らすことが出来る。

30

【0069】

従って、指数テーブル (32) の出力は入力値に於ける指数関数の値を 8 bit の固定小数点数で表わしたのから 0.5 即ち 8 bit の固定小数点数 0.1000001 を引いた 6 bit の固定小数点数とし、このとき出力レンジは $0 \sim 0.5$ (厳密には 0.5 は含まない) である。

【0070】

指数テーブル (32) も前記対数テーブル (12) と同様、RAM や ROM で作り、入力値をアドレスに変換して参照するように構成することもできるが、ここでは、出力論理値を入力論理値の論理式で表現して論理式に対応する回路で指数テーブル (32) を構成する。実際に論理合成してみた結果、 $0.35 \mu\text{m}$ の CMOS で約 1 k ゲートを要した。

40

【0071】

最後に図 6 を使って指数シフト部 (33) の動作を説明する。指数シフト部 (33) の入力は減算部の出力である減算数と指数テーブル (32) の出力である。上述したように、指数テーブル (32) の出力は入力値に於ける指数関数の値を 8 bit の固定小数点数で表わしたのから 0.5 即ち 8 bit の固定小数点数 0.1000001 を引いた 6 bit の固定小数点数であるから、指数シフト部 (33) は逆に指数テーブル (32) の出力に 0.5 即ち 8 bit

50

の固定小数点数0.1000001を足して、値域を0.5～1（厳密には0.5は含まない）に戻す必要がある。次にその値を減算量だけ右シフトして出力する。

【0072】

（a）の場合、指数テーブル（32）の出力0.01011に8bitの固定小数点数0.1000001を足して、減算量2だけ右シフトすると、出力値0.0010011を得る。但し、右シフトで上位bitが空いたところには0が入る。

【0073】

（b）の場合、指数テーブル（32）の出力1.01101に8bitの固定小数点数0.1000001を足して、減算量5だけ右シフトすると、出力値0.0000011を得る。

【0074】

上述のように動作する指数シフト部（33）の回路図を図7に示す。指数シフト部の入力は指数減算部からの出力である減算量3bitと指数テーブル（32）からの出力6bitである。指数テーブル（32）からの出力に対しては入力直後に8bitの固定小数点数0.1000001を足し算しておく。該足し算結果は8bitの固定小数点数である。

【0075】

シフトに関する論理は大きく3段に分かれる。まず、1段目では減算数の最下位が1のとき、該足し算結果を1bit右シフトし、減算数の最下位が0のとき、該足し算結果を右シフトしない。

【0076】

次に、2段目では減算数の2桁目が1のとき、該足し算結果の1段目におけるシフト結果を2bit右シフトし、減算数の2桁目が0のとき、該足し算結果の1段目におけるシフト結果を右シフトしない。

【0077】

最後に、3段目では減算数の最上位が1のとき、該足し算結果の2段目におけるシフト結果を4bit右シフトし、減算数の最上位が0のとき、該足し算結果の2段目におけるシフト結果を右シフトしない。

【0078】

本実施例ではべき乗計算部全てを0.35μmのCMOSに実装した場合、約7.5kゲートを要し、約35nsecで演算が完了する。これによって、光源計算をGPIF（0000）チップの中に埋め込むことが可能となり、ボトルネックになっているジオメトリプロセッサ（5000）の処理を軽減することができた結果、システムとして約2倍性能を向上することができた。

【0079】

【発明の効果】

以上、詳細に説明したように、本発明のデジタルべき乗演算装置はテーブル参照によって演算を行うため、ループ計算より高速に演算結果を得ることができる。

【0080】

また、対数テーブルと指数テーブルとの2つにテーブルを分割することで、各テーブルの入力を1つにできて、テーブルの容量を減らすことができる。

【0081】

また、前記対数算出部の入力値が前記対数テーブルの入力値域に含まれない場合に該入力値に適当な整数Lに対し 2^L を乗算し、該乗算結果を前記対数テーブルの入力とし対数テーブル参照後、該参照値にLを加算することで更に対数テーブルの容量を減らすことができ、

前記指数算出部の入力値が前記指数テーブルの入力値域に含まれない場合に該入力値から適当な整数Mを減算し、該減算結果を前記指数テーブルの入力とし指数テーブル参照後、該参照値に 2^{-M} を乗算することで指数テーブルの容量を減らすことができる。

【図面の簡単な説明】

【図1】 デジタルべき乗演算装置の回路構成を示す図。

【図2】 対数算出部の動作を示す図。

10

20

30

40

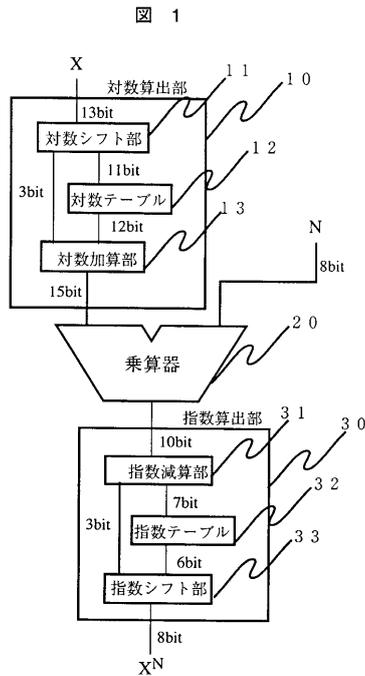
50

- 【図3】対数シフト部の動作を示す図。
- 【図4】対数シフト部の回路構成を示す図。
- 【図5】指数算出部の動作を示す図。
- 【図6】指数シフト部の動作を示す図。
- 【図7】指数シフト部の回路構成を示す図。
- 【図8】グラフィックス・システムの構成を示す図。
- 【図9】光源テーブル及び光源計算手段の構成を示す図。
- 【符号の説明】

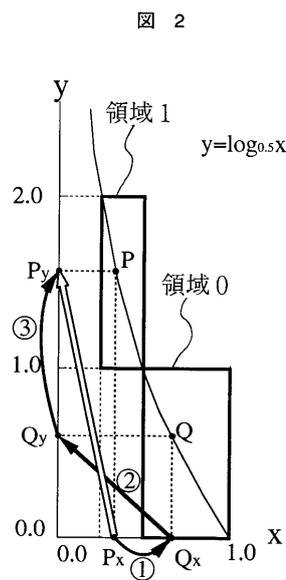
00...べき乗算出部、10...対数算出部、11...対数シフト部、12...対数テーブル、13...対数加算部、20...乗算器、30...指数算出部、31...指数減算部、32...指数テーブル、33...指数シフト部、000...光源計算手段、010...HN内積算出部、020...LN内積算出部、030...色算出部、100...GPIF入力手段、200...LBuf、300...BufSW、400...FI変換手段、500...パック手段、600...コマンド解釈手段、700...光源テーブル、800...制御手段、900...CBuf、950...BufFL、0000...GPIF、1000...CPU、2000...MC、3000...MM、4000...システムバス・コントローラ、5000...ジオメトリプロセッサ、6000...レンダリングプロセッサ、7000...フレームメモリ、8000...CRT。

10

【図1】



【図2】



フロントページの続き

- (72)発明者 鈴木 克徳
茨城県日立市大みか町七丁目1番1号 株式会社 日立製作所 日立研究所内
- (72)発明者 高見 和久
茨城県日立市大みか町七丁目1番1号 株式会社 日立製作所 日立研究所内
- (72)発明者 鬼木 一徳
茨城県日立市大みか町五丁目2番1号 株式会社 日立製作所 大みか工場内

合議体

- 審判長 川崎 健
審判官 右田 勝則
審判官 和田 志郎

- (56)参考文献 特開昭62-257529(JP,A)
国際公開第96/28774(WO,A1)

- (58)調査した分野(Int.Cl., DB名)
G06F7/522