C-011

高基数選択型浮動小数点数除算器の高速化
Fast method of a high radix selection type floating point divider

石田　泰明†*，森下　賢幸†，小椋　清孝†，伊藤　信之†
Yasuaki Ishida,　Takayuki Morishita,　Kiyotaka Komoku,　Nobuyuki Itoh

1. 研究背景・目的

近年の 3D 画像処理等において，浮動小数点数除算器がよく用いられる．しかし，除算は他の演算と比較して多くの処理時間を要すため，除算器の高速化が用いられている．そこで，過去に IEEE754 方式の 32bit の単精度浮動小数点数に対応した，基数 16 の高基数選択型浮動小数点数除算器を FPGA 上にて設計した[1]．しかし，処理時間・回路規模共に課題を残した．

そこで本研究では，処理時間に比重を置いて改善すべく，前処理のアルゴリズムを見直し，乗算手法・正規化の最適化を行った．

2. 浮動小数点数除算

浮動小数点数除算は符号部・指数部・仮数部のそれぞれにおいて商を求める必要がある．符号部の商をQ_s，指数部の商をQ_e，仮数部の商をQ_mとすると以下にて求まる．

$$\boldsymbol{Q_s = N_s exor D_s} \quad (1)$$
$$\boldsymbol{Q_e = N_e - D_e + 127} \quad (2)$$
$$\boldsymbol{Q_m = N_m \div D_m} \quad (3)$$

ここで，N_e, D_eは被除数，除数の指数部を，N_s, D_sはそれぞれの符号部を，N_m, D_mはそれぞれの仮数部を示す．以上の式を見ると，式(3)の仮数部の商Q_mの導出に除算が必要であり，時間が掛かることが分かる．したがって，仮数部の計算の高速化が課題である．

3. 高基数除算法[2]

本研究で用いる除算アルゴリズムとして，高基数除算法を採用する．

$$\boldsymbol{R_i = rR_{i-1} - q_i D_m} \quad (4)$$

ここで，rは基数，q_iは$(\log_2 r)$bit の商，R_iは部分剰余，D_mは仮数部の除数である．人が行う筆算の様に式(4)を繰り返し，その都度q_iを$(\log_2 r)$bit ずつ確定させていくことで最終的に仮数部の 24bit の商Q_mを求める．つまり，基数を増加させることで 1 サイクルにて求まる部分商が増加するため，サイクル数の低減に繋がる．尚，部分商q_iは以下の式(5)と式(4)を満たすq_iを選択する．

$$\boldsymbol{|R_i| \le D_m} \quad (5)$$

式(4)，(5)と，基数は 16 を用いることから式(6)へと変換できる．

$$\frac{\boldsymbol{(q_i - 1)D_m}}{\boldsymbol{16}} \le \boldsymbol{R_{i-1}} \le \frac{\boldsymbol{(q_i + 1)D_m}}{\boldsymbol{16}} \quad (6)$$

商選択を行うため，式(6)をグラフ化したものを P-D プロットと呼び，図 1 にて示す．例としてR_iとD_mの関係が図 1 の×印であったとすると，式(6)を満たす領域は図 1 における赤の領域であり，q_iは 3 と選択される．

尚，ここにはスケーリングを適用している．これは，ある数をスケーリング係数 S を乗算することで，任意の範囲に収束させる演算である．結果，図 1 に示したプロット範囲の本来必要であった青色の領域を削減して赤色の領域のみで商選択を行えるため，デコーダの高速化・小型化に繋がる．

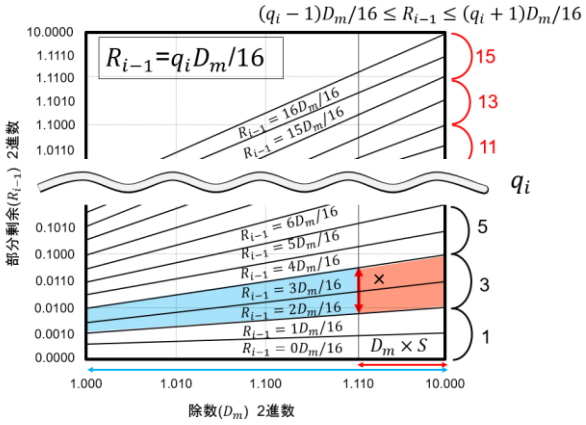

図 1 P-D プロット(基数 16)

4. 選択型回路[3]

高基数除算法の問題点として挙げられるのが，R_iとq_iが依存関係にあることにより，回路が逐次処理となることであった．これを改善すべく提案されたものが「選択型」という手法であり，図 2 にアルゴリズムを示す．具体的には，q_iの算出が終了すればR_iとq_{i+1}の計算に移るが，これと並列してq_{i+1}が確定する前にR_{i+1}を可能性のあるすべてのq_{i+1}の候補に対して計算する．具体的には，基数 16 であれば 16 通りのq_{i+1}が存在するため，合計 16 個の加減算器を並列に計算させる．したがって，q_{i+1}が算出されれば，計算済みのq_{i+1}から対応するものを選択するだけでR_{i+1}を求めることができる．

以上より，本来逐次処理であれば必要であった，図 3 下部の薄い線にて示した加減算の処理を簡略化でき，1 サイクルで 2 組のR_iとq_iを求めることができるため高速化に繋がる．サイクル数としては，1 度にq_iを 4bit ずつ選択することから，$24 \div (4 \times 2) = 3$ より 3 クロックで計算できる．

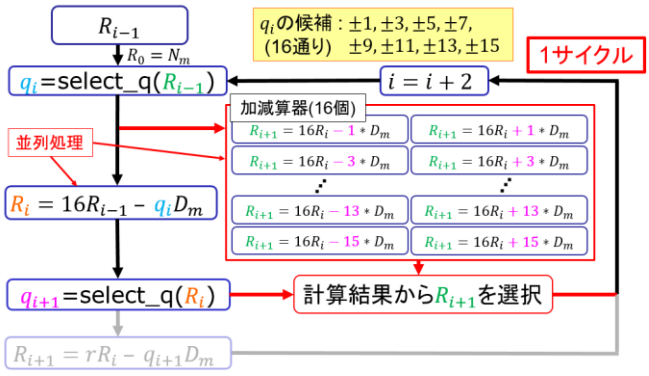

図 2　選択型回路

5. 設計仕様・評価法

全体回路は全 5 クロックで動作する設計仕様としている．1 クロック目は前処理として，被除数・除数の入力から，そのスケーリングと乗算，式(4)にて用いる$q_i D_m$にあたる値の計算・レジスタへの格納を行う．また，符号部・指数部の計算もここで行う．2，3，4 クロックでは選択型回路を用いて仮数部の商を計算し，5 クロック目で正規化・丸め処理を行い，32bit の単精度浮動小数点数を出力する．

以上の回路を Xilinx 社の FPGA 開発環境 ISE 上にて，Verilog HDL を用いて設計した．ターゲットデバイスは Xilinx Virtex-5 である．

評価については，処理時間は同社の Isim 14.7 の Post-route simulation にて，回路規模は ISE の Design Summary より Look Up Table の数を参照する．

6. 回路改善案

従来の回路[1]で遅延の原因となった前処理の回路を改善する．

文献[1]で採用していた前処理のアルゴリズムは図 3 に示した 5 ステップである．ここに並列処理を用いて 4 ステップとしたものが図 4 である．尚，それぞれの仮数部の除数D_mの 1 倍～15 倍の計算については，$x$bit 左にシフトすることで2^x倍を計算できる浮動小数点数の特性を利用して，加減算のみで行っている．

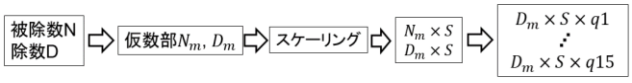

図 3　前処理のアルゴリズム[1]

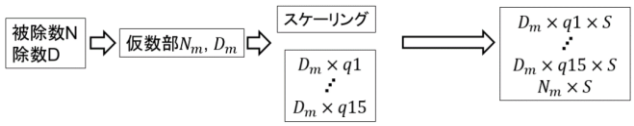

図 4　変更後

スケーリング係数 S の乗算は文献[1]では乗算器を使用していたが，S は 7 通りしか存在しないことから，条件分岐による加算処理へと変更した．ここでは，先程と対照的に，$x$bit 右にシフトすることで2^{-x}倍を計算できる特性を利用している．例として$S = 1.101_2$のときは式(7)のように計算する．

$$D_m \times 1.101_2 = D_m + D_m \times 2^{-1} + D_m \times 2^{-3} \quad (7)$$

これを 8 通りの$D_m \times S$について適用する．

最後に正規化の処理回路を，レジスタを最小限のものとしてクロックを削除することで最適化を行った．

以上の変更を加えた際に評価した処理速度・回路規模を表 1 に示す．

表 1　処理時間・回路規模まとめ

処理時間

	従来方式（逐次型）①	回路[1] ②	アルゴリズム改善後③	加算条件分岐 ④	正規化改善後 ⑤
基数8	88.4ns	64.5ns		59.9ns	
基数16	69.5ns	98.5ns	75.7ns	52.8ns	49.2ns

回路規模

	従来方式（逐次型）①	回路[1] ②	アルゴリズム改善後③	加算条件分岐 ④	正規化改善後 ⑤
基数8	922	953		1312	
基数16	1576	1422	2236	2199	2155

7. 従来方式との比較

基数 8 の従来方式の逐次型回路を基準(図 5 中赤線)として，それぞれに変更を加えた際の処理速度と回路規模の推移を図 5 に示す．尚，図中の①~⑤は表 1 の番号と対応させている．この図では，上に行くほど速度が速く，右に行くほど回路規模が増大する．

結果，従来方式の基数 8 と比較して，処理時間は約 1.8 倍の高速化，回路規模は約 2.3 倍の増大となった．

基数 16 として比較すると，処理時間は約 1.4 の高速化，回路規模は約 1.5 倍の増大となった．

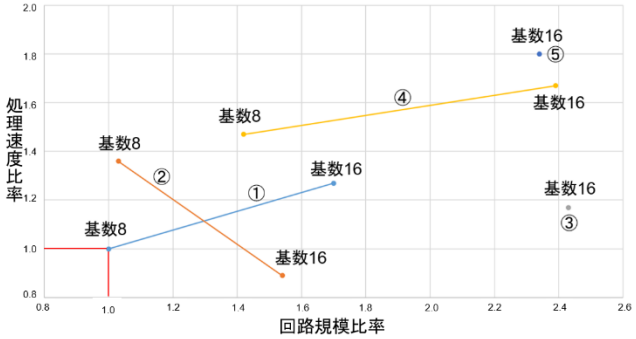

図 5　処理速度-回路規模比率

8. まとめ

本研究では文献[1]にて開発した基数 16 の高基数選択型浮動小数点数除算器の高速化を行った．主に前処理の回路のアルゴリズム・乗算方式の変更によって，従来方式の逐次型回路(基数 8)と比較して，処理時間は約 1.8 倍の高速化，回路規模は約 2.3 倍の増加となった．

今後の課題として，他除算法との比較評価を行うため，同一の FPGA 上にて他除算法の回路をハードウェア化すること，及び更なる数値向上を目指し，選択型回路の改良に取り組むことが挙げられる．

参考文献

[1] 石田　泰明，森下　賢幸，小椋　清孝，伊藤　信之，“高基数選択型浮動小数点数除算器の開発”，平成 30 年度(第 69 回)電気・情報関連学会中国支部連合大会，2018/10/20

[2] 榎本忠儀，堀口智哉，”高基数・スケーリング方式浮動小数点数除算回路の高速化と小形化”，電子情報通信学会論文誌 C Vol.J86-C, No.8, pp.853-856, 2003

[3] 田邊　寛，森下　賢幸，小椋　清孝，伊藤　信之，”基数 8 を用いた選択型・スケーリング方式浮動小数点数除算器”，　平成 27 年度 (第 66 回) 電気・情報関連学会中国支部連合大会，12-1，宇部，2015 年 10 月 17 日．

† 岡山県立大学大学院情報系工学研究科
GRADUATE SCHOOL OF COMPUTER SCIENCE AND SYSTEMS ENGINEERING,OKATYAMA PREFECTUAL UNIVERSITY