

FPGA に適した浮動小数点演算器の構成法

李 寧† 富岡 洋一† 宮崎 昭彦‡ 北澤 仁志†
 †東京農工大学 ‡NTT マイクロシステムインテグレーション研究所

1 はじめに

近年、FPGA の急速な進展により、FPGA による並列処理の研究が活発に行なわれている。その中の多くの研究では固定小数点演算器が使われているため、予めレンジの予測が必要となる [1]。今後高精度化と実用性を高めるには FPGA による浮動小数点演算の並列実行が重要になってくる。本研究では SIMD array による FDTD 法の並列実行を対象とした浮動小数点演算器の構成法を提案し、Altera 社が提供しているメガファンクションや free の浮動小数点演算器合成ツール FloPoCo[2] に比較して、チップ内により多くの PE を実装可能であることを報告する。

2 IEEE754 規格

IEEE754 は浮動小数点数値表現の規格の中で最も使われている規格である。精度によっていくつの形式を定めているが、本報告では単精度の演算について論じる (図 1)。

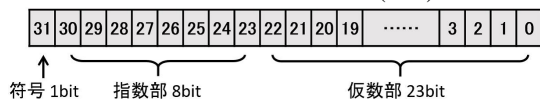


図 1 IEEE754 規格の単精度のフォーマット

指数部と仮数部の値によって、単精度の場合の例外は表 1 のように定義されている。

表 1 IEEE754 規格の例外についての定義

単精度		定義
指数部 (+127)	仮数部	
0	0	0
0	≠0	± 非正規化数
1~254	任意	正規化数
255	0	± 無限大
255	≠0	非数

3 既存手法

3.1 Altera Megafunction Library

FPGA メーカーの Altera 社が提供している library である。Altera の浮動小数点演算器は非正規化数に対応しない点以外、IEEE754 に準拠している。また、丸めモードは一番近い値に丸めるモードになっている。資源は極めて大きい。

3.2 FloPoCo

FloPoCo(Floating-Point Cores) は演算器を自動生成する無料のツールである。コマンドで演算器の種類、bit 長と実装するデバイス、動作速度を入力すると、VHDL コードの演算器

が自動生成できる。IEEE754 形式に 2bit を加え 34bit とし、拡張 bit により 0、正規化数、± 無限大、非数を判定する方法で、Altera と同様に非正規化数以外の例外を IEEE754 に準拠して処理できる。拡張 bit を用いる方法では仮数部と指数部のチェックなどの処理が簡単になるため、Altera の浮動小数点演算器より省資源である。しかし、拡張 bit によって、データパスは 34bit に合わせる必要があり、また、メモリに保存する時、より多くの空間を使う。FloPoCo の演算器も一番近い値に丸めるモードをとっている。

4 提案手法

4.1 浮動小数点演算器の機能の限定

IEEE754 はあくまで汎用の規格であり、FPGA への実装にあたって、その規格に完全に準拠する必要はない。本報告では FPGA の柔軟性を活かして、目的に合わせて浮動小数点演算器を設計する手法を示す。まず、FDTD 法の演算に合わせて、演算器が扱う数値を 0 と正規化数のみに限定した (表 1 のグレーの部分)。それ以外のケースを例外とせず、正規化数として扱う。次に、丸めモードについて、一番近い値に丸めるには最低でも仮数部と指数部の再調整用にもう一段の加算器が必要である。資源の使用量が増えることに加えて、回路が複雑になって、動作速度にも影響がある。我々はより多くの PE を実装することと 100MHz 以上の動作速度を最優先に考えて、IEEE754 規格の中の切り捨モードを選んだ。

4.2 DSP によるシフト演算

近年の FPGA には高性能の DSP ブロックが多数搭載されている。本研究で使用した FPGA は Stratix V の 5SGSMD5K2F40C2 であり、その中には可変精度の DSP が 1590 個内蔵されている。DSP 中の乗算器は一般的には乗算でしか使わないが、提案手法では加減算に必要な Pre-Shift と Post-Shift の処理を DSP の乗算器によって実現した (図 2)。ロジックエレメントの一部を DSP に置き換えることで、ロジックエレメント資源が限られている FPGA チップでも、多くの浮動小数点加減算器を実装することができる。

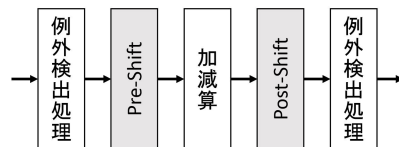


図 2 浮動小数点加減算の手順

4.2.1 Pre-Shift

入力の浮動小数点数を加減算する前に、指数を揃えるため、小さいほうの数値を右に shift する必要がある。一般的に、この部分は 5 段の Barrel Shifter によって構成される (図 3)。提案手法では binary の shift 量を one-hot の形式に変換して、

Implementation of floating point arithmetic units for FPGAs
 †Ning LI †Yoichi TOMIOKA †Hitoshi KITAZAWA
 ‡Akihiko MIYAZAKI
 †Tokyo University of Agriculture and Technology
 ‡NTT Microsystem Integration Laboratories

DSP で上位と下位の bit を入れ替えた被 shift 数値と乗算して、最後に上位と下位の bit を元に戻す方法で Pre-Shift を実現している (図 4)。

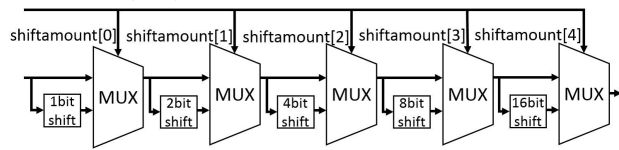


図 3 Barrel Shifter による Pre-Shift

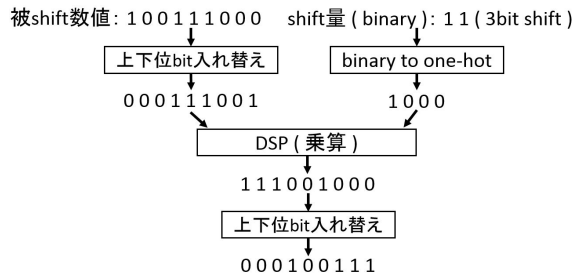


図 4 DSP による Pre-Shift の処理例

4.2.2 Post-Shift

加減算を実行した後、Leading One Detector(LOD)によって、上位から最初に 1 になった桁を探し、それを最上位になるような shift 量を出力する。そして、Post-Shifter はその shift 量に従って、加減算結果を左に shift する。一般的に、LOD は Priority Encoder によって構成され、Post-Shifter は図 3 と似たような 5 段の Barrel Shifter によって構成される。

提案手法では、文献 [3] の Bitscan(Priority Masking) に基づいて LOD を作成した。図 5 に示すように、論理と数値の演算によって、直接 one-hot 形式の shift 量を出力することができる。

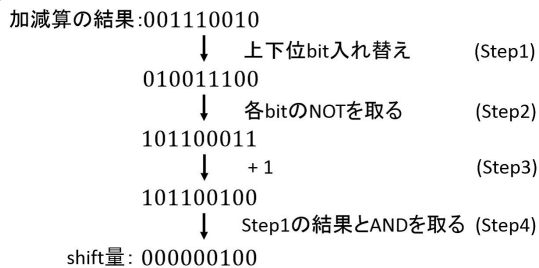


図 5 提案手法の Leading One Detector の処理例

LOD の出力を DSP で加減算の結果と乗算して、Post-Shift を実現する (図 6)。ただし、指数部の再調整には binary 形式の shift 量が必要のため、one-hot 形式の shift 量は Decoder によって binay に変換される。

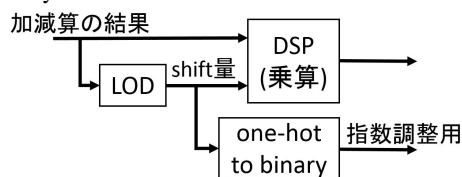


図 6 DSP による Post-Shift

5 実装結果

今回提案している浮動小数点演算器を Verilog HDL で記述し、5SGSMD5K2F40C2 を実装デバイスとし、Quartus II 13.0sp1(balance 重視) の環境でコンパイルした。表 2 にその資源と動作速度を示す。同時に Altera と FloPoCo の合成結果も示す。lut の使用量について、提案手法の乗算器を既存手法と比較すると、Altera の 32%, FloPoCo の 76% に減少している。

表 2 既存手法と提案手法による浮動小数点演算器の比較

演算器の種類	lut	reg	DSP	レイテンシ	動作速度
Altera 標準乗算器	109	191	1	5 clock	463.61 MHz
FloPoCo 乗算器	46	0	2	1 clock ¹	212.09 MHz
提案の乗算器	35	0	1	1 clock ¹	209.86 MHz
Altera 標準加減算器	343	398	0	7 clock	430.85 MHz
FloPoCo 加減算器	282	85	0	1 clock	158.98 MHz
提案の加減算器	155	34	2	1 clock	132.40 MHz

¹ 組み合わせ回路

この結果より浮動小数点演算器の機能の限定にも lut の削減の効果はあるが、主たる要因は shifter として DSP を利用したこと、LOD など周囲の回路も DSP の使い方に合わせて作成したためである。

既存手法による浮動小数点演算器と提案手法の浮動小数点演算器を文献 [1] の Processing Element(PE) に実装して、area 重視でコンパイルした結果を表 3 に示す。ただし、FloPoCo の最大実装 PE 数は回路全体のデータパスを 32bit のままで、演算器の拡張 bit を正規化数に固定してコンパイルした結果であり、正確な最大実装 PE 数はここで示した数値より小さい値になると思われる。

表 3 提案手法による浮動小数点演算器の資源と動作速度

演算器の種類	lut(%)	DSP(%)	最大実装 PE 数	動作速度
Altera 標準乗算器	90%	10	156	113.10 MHz
FloPoCo 乗算器	93%	26	210	102.99 MHz
提案の演算器	94%	81	256	102.09 MHz

6 おわりに

本報告では使用目的に応じて機能を限定し、DSP による shift 演算を活用することで、Altera の演算器を使う場合の約 1.6 倍、FloPoCo の演算器を使う場合の約 1.2 倍の PE を実装できることを示した。今後はこの浮動小数点演算器を活用し、FPGA を用いた並列処理において固定小数点で実現できなかった大きな変域の数値計算を実現していく予定である。

参考文献

[1] 高須涼太, 長谷川天平, 富岡洋一, 柴田随道, 中西衛, 北澤仁志, FPGA による FDTD 法の並列実行における固定小数点演算と浮動小数点演算の比較, 信学技報, vol. 113, no. 26, EST2013-9, pp. 45-50, 2013 年 5 月。
 [2] “Welcome to the FloPoCo project”, <http://flopoco.gforge.inria.fr/>
 [3] Altera, Advanced Synthesis Cookbook, Chapter 6-1 Bitscan(Priority Masking), p.41.