

# 任意精度演算器の SFL 記述の自動生成に関する研究

佐藤 佑己<sup>†</sup> 渡邊 誠也<sup>†</sup> 名古屋 彰<sup>†</sup>  
<sup>†</sup> 岡山大学大学院自然科学研究科

## 1. はじめに

近年では、FPGA などのハードウェアを用いて様々なシステムの高速化を図る試みが行われている。特に、浮動小数点演算のような複雑な演算では、ソフトウェア実装と比べて低消費電力化や処理の高速化などが期待できる。また、ハードウェアでは IEEE754 の形式にとらわれることなく自由に演算の精度を設定することが可能である。

しかし、アプリケーションをハードウェアで実装する場合、設計者にかかる負担はソフトウェア実装と比べて大きい。

そこで本研究では、SFL 記述の任意精度浮動小数点演算器を自動生成するツールを提案し、実装する。これにより、SFL を用いたハードウェア設計の工数を削減し、設計者の負担の軽減を試みる。

## 2. 設計方針

提案するツールを用いてハードウェア設計を行う場合の設計フローを図 1 に示す。設計者が必要とする精度と演算器の種類を指定することによって、指定された精度の演算器の SFL 記述を生成するツールを実装する。このツールは C 言語を用いて実装する。加算器や乗算器などの基本的な演算器については、IEEE754 形式の演算に準拠したものとす。

演算器の内部で行われる加減算や乗算について、オペレータを用いる記述と、演算器を用いる記述を選択できる機能を設ける。オペレータを用いる場合、生成された SFL 記述は SFL Compiler[1]で Verilog HDL 記述に変換して論理合成を行う。オペレータを使用せずに演算器のみを用いる場合は、PARTHENON[2]の合成系が利用できる。内部で生成される演算器の構成についても任意に指定可能とする。

また、指定する段数でパイプライン化された演算器の SFL 記述を生成する機能を設ける。演算器の処理の区切りにパイプラインレジスタを挿入、あるいは内部の演算器自体のパイプライン化により実現する。

## 3. 評価方法

本研究で提案したツールの動作検証には、任意精度の浮動小数点演算器生成プログラムである FloPoCo[3]を用いる。生成された演算器が正常に動作していることを確認できれば、それぞれの演算器に対して論理合成を行い、性能を評価する。

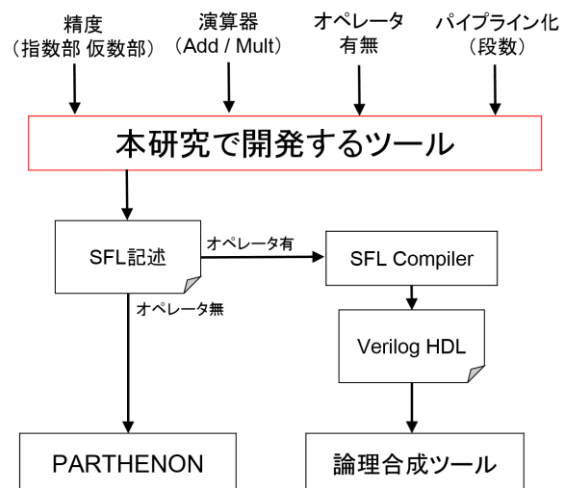


図 1. 提案するツールを用いた場合の設計フロー

表 1. 浮動小数点加算器の論理合成結果

精度	ALM 数	最大動作周波数
16 ビット	178	54.65 MHz
32 ビット	325	37.79 MHz

さらに、浮動小数点演算を利用するアプリケーションを本ツールにより得られた演算器を用いて SFL で記述し、FPGA に実装を行う。同様の処理を行うアプリケーションを C 言語でソフトウェア実装を行い、両者の実行時間や消費電力について比較する。

## 4. 現状と今後

現状では、IEEE754 形式の半精度と単精度の浮動小数点加算器を SFL で実装した。SFL Compiler によって Verilog HDL 記述に変換し、Quartus Prime を用いて Cyclone V 向けに論理合成した結果を表 1 に示す。これらの記述を元にして、任意精度浮動小数点加算器を生成するツールを実装し、評価を行う。

その後は、浮動小数点乗算器、オペレータを用いない SFL 記述、パイプライン化機能の追加を順次行う。

## 参考文献

- [1] 渡邊 誠也, 名古屋 彰, "ハードウェア記述言語 SFL から Verilog HDL への変換系「SFL コンパイラ」の設計と実装," 第 41 回パルテノン研究会資料集, pp. 1-8, 2015.
- [2] <http://www.parthenon-society.com/archive/NTT/>.
- [3] <http://flopoco.gforge.inria.fr/>.