

Chisel を用いたハードウェア設計の 効率化に関する研究

泉倉 大地[†] 渡邊 誠也[†] 名古屋 彰[†]

[†] 岡山大学大学院自然科学研究科

1. はじめに

設計対象となるハードウェアが大規模になるにつれ、効率的なハードウェア設計手法が求められている。

また、近年のハードウェア技術の発展により、プロセッサの性能は年々向上しており、プロセッサの開発やプロセッサアーキテクチャの研究では膨大なコストや開発期間が要する。これらを削減するため、製造前の段階でプロセッサの動作や性能をシミュレーションすることは必要不可欠となっている。

そこで、本研究では、プログラミング言語 Scala[1]と、ハードウェア設計のための Scala のドメイン固有言語である Chisel[2]を利用し、Chisel で設計したハードウェアのシミュレーションに関する処理の効率化・高速化を行うことで、ハードウェアの研究・開発の効率化に寄与することを目指す。

2. Chisel のシミュレーションにおける動作概要

Chisel は Scala の内部 DSL として実装されたオープンソースのハードウェア記述言語である。設計対象が count4 という名称の場合の Chisel を用いたシミュレーションの流れを図 1 に示す。

まず、Scala の内部 DSL で設計対象のハードウェアを記述したファイル(count4.scala)を作成し、Scala コンパイラでコンパイルし、クラスファイル(class file)を生成する。その後、生成されたクラスファイルと Chisel で提供されているクラスファイルを用いてプログラムを実行することにより、Verilog HDL 記述(count4.v)または、C++のソフトウェアシミュレータ(count4)を出力することが可能である。

3. シミュレーションの性能向上手法

Chisel を用いたハードウェア設計のうち、シミュレーションの性能向上手法として、以下の 3 種類の手法を実装した。

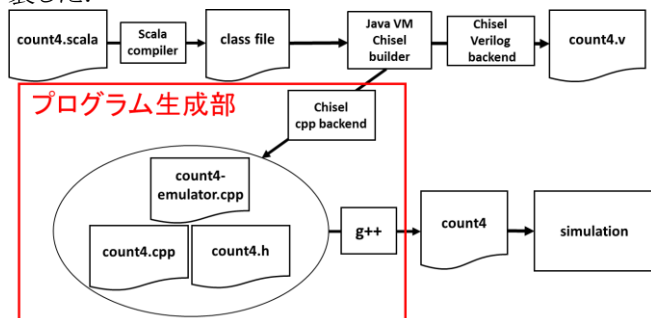


図1. Chisel を用いたシミュレーションの流れ

(1) プログラム生成部の効率化

Chisel ではシミュレーションを行うために、C++で記述されたコードを生成し、コンパイルを行うが、大規模なハードウェアを設計する際、そのコンパイル時間が長大になる場合がある。この問題を解決するために、従来単独でコンパイルを行ってきたものを、並列にコンパイルすることにより、コンパイル時間の短縮を図った。

(2) シミュレーションの高速化

Chisel によって設計したハードウェアの動作、性能を検証するシミュレーション処理の高速化を行う。シミュレーションを行う C++で記述されたコードの関数に着目し、関数毎に変更を行い、全体のシミュレーション時間の短縮を図った。

(3) スループットの向上

多種多様なデータを入力し、その動作、性能を検証することは、より良いハードウェアの設計を行う際には必要な工程である。そこで、単一のシミュレーションワークロードの処理時間を短縮するのではなく、複数のシミュレーションワークロードをマルチプロセスで実行する際の処理の効率化を図り、スループットの向上を目指した。

4. 評価

提案する効率化手法を実装し、オリジナルの Chisel と比較評価を行った。プログラム生成部の効率化手法では最大で約 9.5 倍の性能となった。一方、シミュレーションの高速化手法では約 1.07 倍の性能に留まった。スループット向上手法ではマルチコアマシン(物理 8 コア, 論理 16 コア)を利用し、オリジナルの Chisel の逐次実行に比べて、単なるマルチプロセス実行では約 6.2 倍の性能であるのに対して、本研究の効率化により最大で約 12.2 倍の性能となった。

5. 今後の課題

さらなるスループットの向上を実現するためには、Chisel 自身が複数のシミュレーションワークロードを管理する手法の実装を行う必要がある。

参考文献

- [1] The Scala Programming Language, <http://www.scala-lang.org/>.
- [2] Chisel, The Regents of the University of California, <http://chisel.eecs.berkeley.edu/>.