

RISC-V プロセッサにおける整数乗算器の実装

1200018 池野 樹 (集積システム研究室)
(指導教員 密山 幸男 准教授)

1. はじめに

RISC-V という ISA (Instruction Set Architecture) は、モジュール式で開発が行われているため、ターゲット・アプリケーションに応じて命令拡張機能のモジュールをハードウェアに取り込むか否かの選択を自由に行うことができる。この特徴を用いて、特定用途向け回路を混載させることにより、プロセッサの性能向上を計ることができる。本研究では命令拡張機能の 1 つとして、乗算器の設計に注目する。様々な種類の乗算器を基本命令セット (RV32I) の RISC-V コアに追加し、FPGA に実装することで、乗算演算の性能ならびに、リソース使用量や消費電力を比較評価する。

2. ハードウェア

2.1 プロセッサの構成

ISA を RV32I とした RISC-V プロセッサの構成を図 1 に示す。レジスタは 32bit 幅で構成されており、整数のみを扱うことができる。コンパイルされたプログラムはデータメモリと命令メモリもあわせて論理合成される。ALU は算術論理演算ユニットであり、ここに乗算器を実装する。

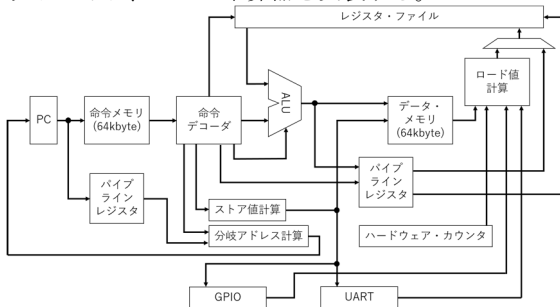


図 1 プロセッサの構成概要図

2.2 乗算器

本研究で用いる 3 種類の乗算器について説明する。1 つ目は Verilog HDL 記述に用いる算術演算子による乗算器である。

2 つ目は図 2 のように部分積を生成した後、リップルキャリー型加算器を用いて全ての段を加算する乗算器である。本研究ではこの乗算器をリップルキャリー型乗算器と呼ぶ。乗数のビット幅が大きくなると段数が多くなるため、加算を行う回数が増え、下位からの桁上げを待たなければならず、遅延が大きくなる。

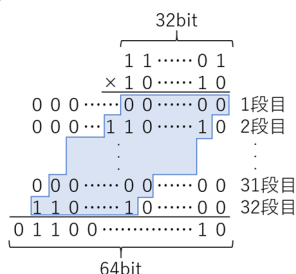


図 2 32bit×32bit の基本的な乗算

3 つ目は Wallace tree を用いた乗算器である。図 3 は乗数を a(8bit)、被乗数を b(8bit)、乗算結果を p(8bit) とした場合の例である。青い箇所が全加算器、緑の箇所では半加算器を用いて加算を行う。Stage0 の時点で部分積を上詰めてから加算を行うことで不必要な加算器を減らすため、リソースの削減や計算速度の向上につながる。

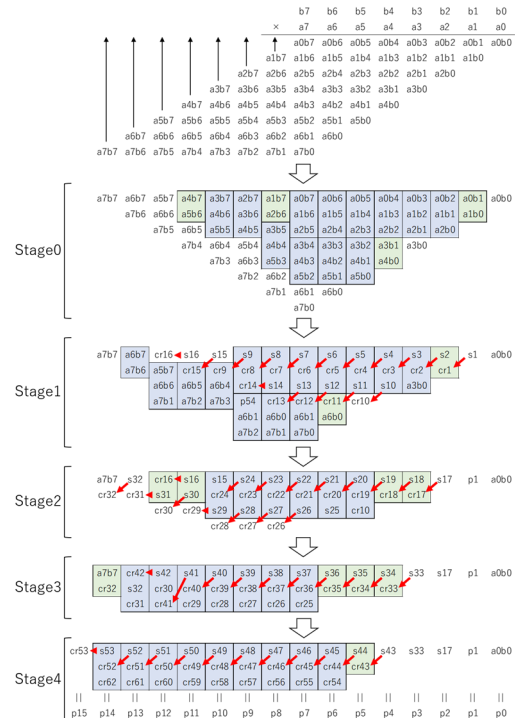


図 3 Wallace tree を用いた部分積加算

3. 実装結果

表 1～表 3 に実装結果を示す。RV32I は乗算器が無い基本命令セットのみ、RV32I+A が算術演算子による乗算、RV32I+R がリップルキャリー型乗算器、RV32I+W が Wallace tree 型乗算器を用いた時の実装結果を表している。消費電力は論理合成による見積り値であるため、乗算器による変化はあまり見られなかった。計算速度は算術演算子を用いた乗算器が他の乗算器と比べて約 1.8 倍高速であり、リップルキャリー型と Wallace 型はほぼ同じであった。

表 1 LUT の使用量

	RV32I	RV32I+W	RV32I+R	RV32I+A
ALU	516	5,258	5,274	6,167

表 2 消費電力

	Total On-chip Power[mW]
RV32I	253
RV32I+W	270
RV32I+R	269
RV32I+A	269

表 3 計算速度

	Time[ps]
RV32I+W	17,865
RV32I+R	18,861
RV32I+A	10,016

4. まとめ

基本命令セット RV32I の RISC-V プロセッサに複数種類の乗算器を搭載し、FPGA を用いた実装結果から比較評価を行った。リップルキャリー型乗算器と Wallace tree 型乗算器の場合で回路規模や計算速度の差が小さくなった原因として、設計した乗算回路に問題があったと考えられる。算術演算子を用いた乗算器は、回路規模は大きくなるが、処理速度は極めて速いことがわかった。

参考文献

- [1] David Patterson and Andrew Waterman, 成田光彰 (訳), “RISC-V 原典”, 日経 BP 社, 2018 年 10 月.
- [2] A. Waterman and K. Asanovic. “The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.2”, May 2017.