

RISC-V へのカスタム命令実装手法の検討

今井 信志[†] 渡邊 誠也[†]
[†] 岡山大学大学院自然科学研究科

須山 敬之^{††} 名古屋 彰[†]
^{††} NTT コミュニケーション科学基礎研究所

1. はじめに

近年, IoT 機器などの低消費電力で動作するデバイスへ搭載するCPUとしてRISC-Vが注目されている. RISC-Vはカリフォルニア大学バークレー校で開発された命令セットアーキテクチャであり, 現在はRISC-V Foundation[1]により開発・管理が行われている.

また, RISC-Vにはカスタム命令用のオペコードが定義されており, その命令の内容は設計者に委ねられている. このカスタム命令を用いれば, 特定領域に特化したRISC-Vを設計することが可能である. しかし, RISC-Vへカスタム命令を実装し, 評価を行った研究は少ない.

そこで, 本研究ではカスタム命令を搭載したRISC-Vをハードウェア記述言語FSL(Functional and Scalable hardware description Language)[2]で設計, FPGA(Field Programmable Gate Array)へ実装し, 論理合成結果とシミュレーションによるベンチマーク結果を用いて評価を行い, カスタム命令実装の指針を得ることを目的とする.

2. 実装するRISC-Vの構成

RISC-Vは命令セットを複数有する[3]. 本研究で実装するRISC-Vはアドレス長を32ビットとし, 実装する命令セットはRV32IMをベースとした.

実装したRISC-Vはインオーダー実行の5段パイプライン構成とし, パイプラインステージは命令フェッチ, 命令デコード, 実行, メモリアクセス, レジスタライトバックとなっている.

3. 実装するカスタム命令

近年, 機械学習分野, 特にニューラルネットワークにおける演算では, 積和演算の高い処理能力が要求される. そこで, 本研究では, RISC-Vへ実装するカスタム命令は積和演算を行うものとした. 実装するカスタム命令を表1へ示す. madd命令, madd44命令ともに演算に用いる数値は32ビット整数とする.

4. 実装手法と現在の実装状況

本研究は, RISC-Vの設計記述言語としてFSLを用いることに特徴がある. FSLは生産性の高いハードウェア設計環境の実現を目指しており, 設計者が意図するハードウェアの動作を適切な抽象度で記述することができる. これにより, 本研究で実施するカスタム命令の追加やパイプライン構成の変更などが容易となる. また, FSLによるRISC-Vの記述例を蓄積しておくことで, 後にハードウェア設計者が用途に応じた構成でRISC-Vを利用したいという際に役立

表 1. 実装するカスタム命令

命令フォーマット	処理内容
madd rd, rs1, rs2	$rd = rd + rs1 * rs2$
madd44 rd	$rd = a0 * a1 + a2 * a3 + a4 * a5 + a6 * a7$

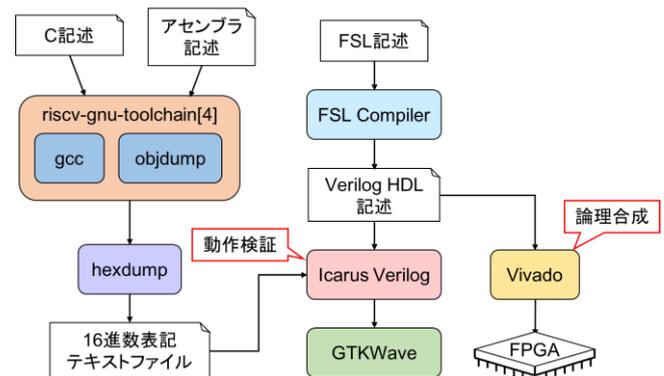


図 1. FSLを用いた設計フロー

表 2. 論理合成結果

LUT数	FF数	DSP数	最大動作周波数 [MHz]	消費電力 [mW]
2,121 (10.20%)	1,432 (3.44%)	10 (11.11%)	51.55	76

()内はArtix-7のリソース使用率

てることもできる.

FSLを用いた設計のフローを図1に示す. FSL記述からFSL Compilerを用いてVerilog HDL記述に変換し, Icarus Verilogを用いてシミュレーションを行うことによって, 動作検証を行う. また, ターゲットデバイスはXilinx社のArtix-7(XC7A35TICSG324-1L)とし, Vivadoを用いて論理合成を行う. 現状ではカスタム命令を搭載していないRISC-Vコアの実装が完了している. 論理合成結果を表2に示す.

5. 今後の予定

今後は2つのカスタム命令をそれぞれ実装し, 論理合成結果と行列積によるベンチマーク結果を用いて評価を行う.

参考文献

- [1] RISC-V Foundation, <https://riscv.org/>.
- [2] 渡邊 誠也, 名古屋 彰, “オブジェクト指向/関数言語をベースとするハードウェア記述言語 FSL の設計,” 電子情報通信学会技術報告, RECONF2015-37, pp. 27-32, 2015.
- [3] A. Waterman and K. Asanovic, The RISC-V Instruction Set Manual Volume I: User-Level ISA Document Version 2.2, RISC-V Foundation, 2017.
- [4] RISC-V Foundation, riscv-gnu-toolchain, <https://github.com/riscv/riscv-gnu-toolchain> (accessed Sep. 2018).