

## 32ビット RISC-V プロセッサ VexRiscv のカスタム命令追加と FPGA 実装

1230038 奥田 真菜 (集積システム研究室)  
(指導教員 密山 幸男 教授)

## 1. はじめに

オープンな命令セットアーキテクチャ (ISA) である RISC-V[1]が注目されており、さまざまな研究開発が行われている。VexRiscv[2]は、RISC-V Soft CPU Contest[3]で1位を獲得したFPGA向け32ビットRISC-Vプロセッサである。本研究では、VexRiscvを対象として高効率プロセッサの開発に向けたカスタム命令追加手法を確立し、プロトタイプの実動作検証まで行う。

## 2. VexRiscv

VexRiscvはFPGAで動作させることを目的として開発されたRISC-Vプロセッサのひとつである。プログラミング言語Scalaのライブラリとして実装されたSpinalHDL[4]を用いて記述されており、VerilogHDLよりソフトウェアに近い記述が可能である。一方で、記述言語であるSpinalHDLがまだ広く使われていないこともあり、情報が限られている。またVexRiscvについてもカスタム命令を追加し、FPGA実装と評価を行った報告はない。

## 3. VexRiscv の FPGA 実装

前章で述べたVexRiscvをFPGA上に実装する。FPGA実装の手順を図1に、開発環境を表1に示す。ターゲットボードにはAvnet社製Ultra96-V2を用いた。Ultra96-V2に搭載されているプログラマブルSoCは、プロセッサコア(PS)とプログラマブルロジック(PL)を搭載している。本研究では、PL上にVexRiscvを実装する。

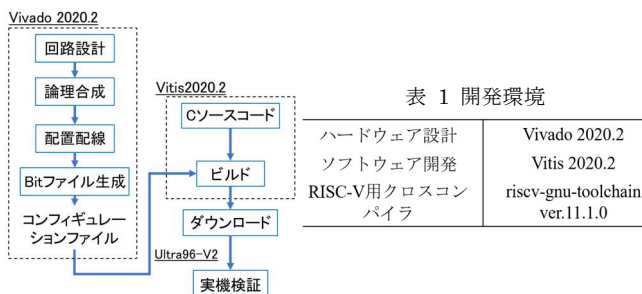


図1 FPGA実装の流れ

SpinalHDLを用いてVexRiscvコアを生成する構成を図2に示す。プラグインによって機能を拡張することができる。

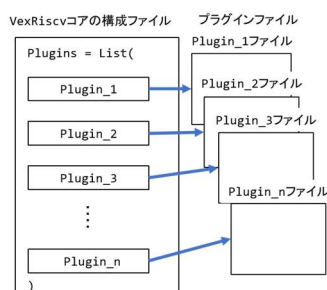


図2 VexRiscv コアの構成

FPGA上に実装したVexRiscvはベアメタルプログラムで動作させる。必要なプログラムは、PSで実行するプログラムとVexRiscvで実行するプログラムの2種類である。

## 4. VexRiscv を用いたカスタム命令の実装方法の提案

評価実験用カスタム命令として、SIMD加算命令、ビット反転命令、パリティ計算命令の3つを実装した。一例として

SIMD加算命令の演算内容を図3に示す。

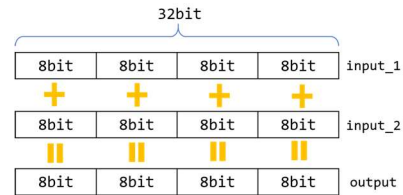


図3 SIMD加算命令の演算内容

## 4.1 ハードウェアの修正

VexRiscvコアに3つのカスタム命令を追加するため、それぞれのプラグインファイルを作成した。例としてSIMD加算命令に関する動作記述を図4に示す。

```
rd(7 downto 0) := rs1(7 downto 0) + rs2(7 downto 0)
rd(16 downto 8) := rs1(16 downto 8) + rs2(16 downto 8)
rd(23 downto 16) := rs1(23 downto 16) + rs2(23 downto 16)
rd(31 downto 24) := rs1(31 downto 24) + rs2(31 downto 24)
```

図4 SIMD加算命令の動作記述

## 4.2 コンパイラの修正

カスタム命令をインラインアセンブリで出力するために、RISC-V用クロスコンパイラを修正する。カスタム命令のMATCH、MASKとDECLAREを定義し、カスタム命令の命令フォーマットを追加する。命令名や命令特定の関数などを定義することができる。例としてSIMD加算命令のインラインアセンブリを図5に示す。

```
asm volatile ("simd_add %0,%1,%2" : "=r" (simd_add)
             : "r" (input_1), "r" (input_2));
```

図5 SIMD加算命令のインラインアセンブリ

## 5. 実装結果

カスタム命令の有無による比較評価を行った。10個の不規則な値を用いて演算を実行した。平均命令数を表2に示す。実験結果から、カスタム命令を追加すると命令数を削減できたことと、評価用のプログラムの逆アセンブル結果と同じ命令数であったことから、カスタム命令を正しく実装できたことを確認した。

表2 カスタム命令実装の結果

カスタム命令の有無	命令数	
	あり	なし
SIMD加算命令	1	16
ビット反転命令	1	23
パリティ計算命令	1	9

## 6. まとめ

本研究では32ビットRISC-VプロセッサVexRiscvを対象として、カスタム命令の追加とそのFPGAを用いたプロトタイプ実装を行った。3つのカスタム命令についてカスタム命令の有無による比較評価を行った結果、カスタム命令が正しく実装されていることを確認した。

## 参考文献

- [1] "RISC-V International," RISC-V International, <https://riscv.org/>, 2021.
- [2] "GitHub - SpinalHDL/VexRiscv: A FPGA friendly 32bit RISC-V CPU implementation," <https://github.com/SpinalHDL/VexRiscv>, 2022.
- [3] "RISC-V SoftCPU Contest," RISC-V International, Oct. 2018. <https://riscv.org/announcements/2018/10/risc-v-contest/>
- [4] "GitHub-SpinalHDL/SpinalHDL:ScalabasedHDL," <https://github.com/SpinalHDL/SpinalHDL>, 2022