

RISC-V 圧縮命令のみを用いた小面積な組み込み用プロセッサに関する研究

田中 陽大 指導教員：佐々木 敬泰

1 はじめに

RISC-V は、近年注目を集めている単純な基本命令と多くの拡張機能を持つオープンな命令セットアーキテクチャである。その拡張機能の一つに圧縮命令がある。この圧縮命令を実装すると、基本命令のみの場合から約 60 % を圧縮命令に置き換えることができ [1]、メモリ効率が向上するため、大容量のメモリを持たない組み込み用としての利用が期待されている。しかし、すべての基本命令を圧縮命令に置き換えることはできないので、圧縮命令を用いる場合も基本命令と併用することを前提としている。そのためデコードの際、現在の命令サイズを調べるハードウェアが追加で必要となり、ハードウェア面積が増加するという問題がある。

もし、すべての命令を圧縮命令のみで表現できればハードウェアを簡素化することができる可能性がある。そこで、本研究では圧縮命令のみからなる命令セットを新たに提案することで、命令サイズを測定するハードウェアの削減や、フェッチ幅の減少から、ハードウェア面積の縮小を目指す。なお、提案する手法は RISC-V の正規の命令セットである RV32IMAC のサブセットのみを用いるためバイナリ互換性が高く、本手法を適用してコンパイルした実行ファイルは既存の RISC-V プロセッサでも実行できるという利点がある。

2 提案手法

RISC-V では圧縮命令と基本命令は混在が前提とされており、圧縮命令のみではプロセッサを動かすには不十分である。そのため、本研究では圧縮命令のみで動作する命令セットの実現方法として、基本命令を複数の既存圧縮命令に対応付けて置き換える方法を提案する。しかし、圧縮命令には OS の動作に必要な特権命令が実装されていないという問題がある。そのため、これらの命令が必要な場合では、本研究で作成するプロセッサはコプロセッサとして動作させることを想定し、汎用プロセッサが特権命令を実行することで対処を行う。

プロセッサを動作させるために必要な 37 個の基本命令を圧縮命令に対応付けた。ここではその一つとして addi 命令の対応を

表 1 addi 命令の圧縮命令対応

条件	基本命令	圧縮命令対応
$im \neq 0 \ \& \ rd=rs1$	addi rd, rd, im	c.addi rd, im
$im \neq 0 \ \& \ rd \neq rs1$	addi rd, src1, im	c.li rd, im c.add rd, src1
$im=0 \ \& \ rd=rs1$	addi rd, rd, 0	c.nop
$im=0 \ \& \ rd \neq rs1$	addi rd, src1, 0	c.mv rd, src1
$\log_2(im) \geq 7$	andi rd, rd, im	c.li rd, im[11:6] c.slli rd, 6 c.addi rd, im[5:0]
$rd=src1=x2$	addi x2, x2, im	c.addi16sp im
$src1=x2$	addi rd, x2, im	c.addi4spn rd, im

表 1 に示す。2 つのレジスタが同じかつ、即値が 0 または 7 ビット以上でないときはそのまま addi 命令を c.addi 命令に置き換えることができるが、示すオペランドによってはそのまま置き換えることができないので、新たに対応方法を考える必要がある。具体的には、圧縮命令では rd レジスタと rs1 レジスタは同じオペランドが指すため、違うレジスタをとることができない。そのため、ほかの圧縮命令である c.li 命令と c.add 命令を用いて対応した。また、即値フィールドの都合から、即値が 0 のときには c.nop 命令や c.mv 命令を用いて対応し、7 ビット以上の即値生成には、即値を上位 6 ビットを生成、6 ビット左シフト、下位 6 ビット生成の 3 つの圧縮命令に分割することで対応した。ただし、頻繁に用いられるスタックポインタ (x2) の演算には、即値フィールドが広い専用の圧縮命令が実装されているため、それを用いて対応を行った。

3 評価

本研究で作成した命令セットを用いて、ソフトプロセッサの設計を行う。ソフトプロセッサの設計として、ノースカロライナ州立大学が提案している RISC-V プロセッサの自動設計ツールである FabScalar-RISCV[2] を用いて行った。しかし、現在の FabScalar-RISCV には圧縮命令が実装されていないため、新たに圧縮命令に対応したデコーダと演算器を作成し、シミュレーションを行った。シミュレーションでは、デコーダに圧縮命令の機械語を入力し、その命令が正しくデコードされていることを確認した。また、演算器にはデコーダからの値を入力することで正しい演算結果が返されることを確認した。これにより、圧縮命令のみで動作するプロセッサが実現可能であることを示した。

4 おわりに

本研究により、圧縮命令のみを用いてバイナリ互換性を保つ命令セットが実現可能であることを示した。また、FabScalar-RISCV を用いて圧縮命令のみを実装したプロセッサを設計し、そのシミュレーションを行うことで動作を確認した。その結果、システムコールや CSR 命令を除いて、圧縮命令のみでの実現が可能になった。またこれらの未対応命令は、汎用プロセッサに実行を任せ、本研究のプロセッサは圧縮命令専用のコプロセッサとして実装することで、システム全体の処理性能向上を図ることができると考えられる。今後の課題として、実際に論理合成を行うことでハードウェア面積を評価し、どの程度フットプリントを削減できたか検証することがあげられる。

参考文献

- [1] A. Waterman, et al.: The RUSC-V Compressed Instruction Set Manual Version 1.9, Technical Report UCB/EECS-2015-209, November 5, 2015,
- [2] R. B. R. Choudhury, et al.: AnyCore: A synthesizable RTL model for exploring and fabricating adaptive superscalar cores, 2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS) pp. 214-224, 2016