

## マイクロプロセッサ rj406 の浮動小数点演算

2C-7

菅原 浩二 青柳 圭祐 大林 雄次 五味 智 清藤 麻子  
電気通信大学 情報工学科

## 概要

我々は RISC マイクロプロセッサ rj406 の設計・製作を進めており、その基本アーキテクチャとシミュレーションについて報告した [1, 2]。今回は、rj406 の浮動小数点演算用コプロセッサのアーキテクチャについて報告する。

## 1 特徴

このコプロセッサは、rj406 に浮動小数点演算機能を提供することを第一の目的としたもので次の特徴をもつ。

- ロード-ストアアーキテクチャ
- データ長 32 ビット
- URR\*[3, 4] を用いた、広い実数の表現範囲
- 固定小数点演算機能
- データレジスタは 32 個

データレジスタ 0(\$f0) は、rj406 同様常に定数 0が入っている。

## 2 データ型

rj406 のコプロセッサが扱うデータ型は、浮動小数点数と固定小数点数(整数)である。

## コプロセッサのデータ型

- 32 ビット URR(浮動小数点)
- 32 ビット 固定小数点数

Floating Point Number Operation of Microprocessor rj406, Kouji Sugawara, Keisuke Aoyagi, Yuji Oobayashi, Satoshi Gomi, Asako Seito, University of Electro-Communications

従来の数値演算コプロセッサでは、通常、数の表現に IEEE 浮動小数点標準表現を用いているが、我々のコプロセッサでは、浮動小数点数の表現に新しい表現として提案されている URR を用いることにした。

## 2.1 URR について

URR は、浮動小数点数の指数部と仮数部の長さを可変長とすることによって表現可能な数の範囲を広くした表現である。1 に近い数では精度が高いという特徴がある。さらに URR で表現された数は、それを固定小数点数としてみたときと順序関係が同じになるので、比較には固定小数点数の比較回路がそのまま使える。

## 2.2 型の変換

浮動小数点演算を行なう際に、固定小数点のデータを浮動小数点数にして演算を行なうことや、その反対に浮動小数点数を整数にすることがある。そういった場合に、型の変換が必要になる。rj406 コプロセッサでは、数の型は 2 種類なので型の変換は簡単である。

コプロセッサで浮動小数点演算のみを行なう場合は、内部では型の変換をする必要はない。したがって、コプロセッサが外部とデータをやりとりする時点で必要に応じて整数 $\leftrightarrow$ 浮動小数点数の型変換ができれば十分である。しかし、このコプロセッサでは、内部で整数演算機能を持つために、内部で整数と浮動小数点数が共存することになる。そのため、内部に型変換命令をもち、データの移動は型に関係なく 1 種類の命令で行なう方法にした。

\*Universal Representation of Real Numbers

表 1: コプロセッサの命令セット

| 命令                                | オペランド         | 説明                                              |
|-----------------------------------|---------------|-------------------------------------------------|
| lwc                               | cd, offset(s) | メモリの内容をレジスタ cd にロードする。                          |
| swc                               | cs, offset(s) | レジスタ cs のデータをメモリに書き込む。                          |
| mtc                               | cd, s         | 汎用レジスタ s の内容をコプロセッサレジスタ cd に転送する。               |
| mfc                               | d, cs         | コプロセッサレジスタ cs の内容を汎用レジスタ d に転送する。               |
| ctc                               | cd, s         | 汎用レジスタ s の内容をコプロセッサの制御レジスタ cd に転送する。            |
| cfc                               | d, cs         | コプロセッサの制御レジスタ cs の内容を汎用レジスタ d に転送する。            |
| cvt.I                             | cd, cs        | コプロセッサレジスタ cs の内容を整数に変換してコプロセッサレジスタ cs に入れる。    |
| cvt.F                             | cd, cs        | コプロセッサレジスタ cs の内容を URR に変換してコプロセッサレジスタ cd に入れる。 |
| mov                               | cd, cs        | データの移動                                          |
| add.F, sub.F, mul.F, div.F, mod.F | cd, cs1, cs2  | 浮動小数点演算                                         |
| add.I, sub.I, mul.I, div.I, mod.I | cd, cs1, cs2  | 整数演算                                            |
| abs.F, abs.I                      | cd, cs        | 絶対値                                             |
| neg.F, neg.I                      | cd, cs        | 算術否定                                            |
| comp.cond                         | cs1, cs2      | cs1 と cs2 を比較してコンディションコードレジスタに結果をセットする。         |
| bct                               | offset        | コンディションコードレジスタの内容が真ならブランチする。                    |
| bcf                               | offset        | コンディションコードレジスタの内容が偽ならブランチする。                    |

### 3 命令セット

コプロセッサの命令セットは上表のようになった。データの移動、演算、比較とその結果による条件分岐命令がある。

rj406 は、整数の乗除算命令をもたないので、コプロセッサに整数の乗除算命令を加えることにした。固定小数点の演算は、乗除算と他の演算を組み合わせで行なわれることになるので、乗除算命令だけ実装するとデータ転送によるオーバーヘッドが大きくなると考えられる。そこで、乗除算の他にも演算命令を加え、コプロセッサで固定小数点演算ができるようにした。

条件分岐は、1 ビットのコンディションコードレジスタを使う方法になっている。これは、rj406 のコプロセッサ分岐命令のフォーマットの制限からレジスタオペランドを記述する余裕がないためである。

比較命令・データ移動命令が共通なので固定小数点演算のために増える命令は、演算命令だけである。変換命令は、型が 2 種類なので相互に変換する 2 命令である。

このように数の表現に URR を用いることとデータ長を 32 ビットのみにするることによって、命令セットは簡潔になった。

### 謝辞

御指導いただいた浜田穂積教授と中川圭介助教授に感謝致します。

### 参考文献

- [1] 清藤ほか, “マイクロプロセッサ RJ406” 情報処理学会第 46 回全国大会 6-9, 1993
- [2] 青柳ほか, “マイクロプロセッサ rj406 のアーキテクチャと評価” 情報処理学会第 47 回全国大会 6-5, 1993
- [3] HAMADA, Hozumi, “A New Real Number Representation and Its Operation” IEEE proceeding of the eighth symposium on computer arithmetic 1987
- [4] 浜田 穂積, “デジタル・システムの数値表現法” bit Vol20, No.3 pp299-315, 1988