マイクロプロセッサ rj406 のアーキテクチャと評価

2G－3

青柳 圭祐　清藤 麻子　菅原 浩二
大林 雄次　五味 智　伊藤 佳夫
電気通信大学

1 概要

rj406[1] は限られたチップ面積で可能な限り大きな処理能力を実現することを目指して設計された 32 ビット RISC マイクロプロセッサである。本稿では rj406 の命令セット、割り込み機能、コプロセッサ命令について述べ、シミュレーションの結果をもとに評価する。

2 命令セット

演算命令は 3 オペランド演算で加減算、論理演算、シフト演算、比較が、レジスタ間とレジスタ-16 ビットイミディエイト間で実行できる。算術演算のイミディエイトは符号拡張される。整数の加減算ではオーバーフローのチェックは行なっていないが、比較命令の結果を用いて補正することができる。

浮動小数点演算は外部のコプロセッサで行なう。

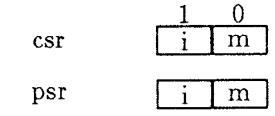
コンディションコードレジスタは持たず、比較命令の結果 (1/0) は、汎用レジスタにはいる。比較命令は slt(set less than) だけであるが、すべての算術比較が 1 ～ 2 命令で実現可能である。

条件つきブランチ命令は、レジスタの内容がゼロかどうか、マイナスかどうかで分岐する 4 命令がある。

3 割り込み

rj406 の割り込みには外部割り込みと trap 命令によるソフトウェア割り込みがある。どちらも割り込みが発生すると 0x8 番地に実行を移す。プロセッサ状態の復帰には reti 命令を jr 命令とともに使用して行なう。

rj406 はプロセッサ状態を保持する ssr(system status register) を 2 段で持つ。システムモードでは ssr のすべてのフィールドに対し読みだし、書き込みが行なえる。csr(current status register) は現在のプロセッサ状態を保持する。割り込みが発生した時は psr(previous status register) にその時のプロセッサ状態が自動的に退避される。また汎用レジスタ 31 番は割り込みの戻り番地退避用に使用される。



m　プロセッサモード。0 のときシステムモード、1 のときユーザモード。
i　割り込みマスク。0 のとき禁止、1 のとき許可。

図 1: システムステータスレジスタ

割り込み関連命令は 6 種ある。trap 命令以外はすべて特権命令で、システムモードでのみ使用可能である。

表 1: 割り込み関連命令

命令	動作
trap	r[31]:=pc; pc:=0x8; psr:=csr; csr:=0
reti rs	csr:=psr; psr:=rs
mtsr ssr,rs	ssr:=rs
mfsr rd,ssr	rd:=ssr
ei	csr[1]:=1 (割り込み許可)
di	csr[1]:=0 (割り込み禁止)

Architecture and evaluation of microprocessor rj406
Keisuke Aoyagi, Asako Seito, Kouji Sugawara,
Yuji Oobayashi, Satoshi Gomi, Yoshio Ito
University of Electro Communications

4 コプロセッサ

rj406 は 4 組のコプロセッサのための命令コード群を持ち、4 個までの密結合型コプロセッサに対応できる。これらのコプロセッサは、浮動小数点演算などの周辺機能を実現する。

コプロセッサ命令は、基本的には rj406 の動作とは独立して解釈と実行が行なわれる。コプロセッサ内部の処理のほかに、コプロセッサのレジスタに対するロード・ストア、rj406 の汎用レジスタとのデータ転送、コプロセッサの条件による分岐が命令として存在するが、これらの機能を実現するためには rj406 とコプロセッサとの協調動作が必要である。

表 2: コプロセッサ命令

命令	動作
lwc$x$ cd,imm(rs)	cd$x$:=m[r[rs]+ext(imm)]
swc$x$ cs,imm(rs)	m[r[rs]+ext(imm)]:=cs$x$
mtc$x$ cd,rs	cd$x$:=r[rs]
mfc$x$ rd,cs	r[rd]:=cs$x$
bc$x$t offset	if (cp_cond=TRUE) pc:=pc+ext(offset)+4
bc$x$f offset	if (cp_cond=FALSE) pc:=pc+ext(offset)+4

この他、各コプロセッサごとの命令セットがある。

cs はコプロセッサのソースを、cd はディスティネーションを示し、x はコプロセッサ番号を示す。r[ ] は rj406 のレジスタである。cs と cd の形式はコプロセッサによって異なる。

5 評価

基本的なプログラムを mips の C コンパイラにかけ、出力コードを rj406 コードに変換し、シミュレータにかけ命令使用頻度の統計をとった。コンパイラ出力コードは mips 特有の命令ではなく、広い範囲の命令で構成されており、rj406 にも類似した形式なので統計をとるために利用した。

使用したプログラムは、ハノイの塔 (hanoi)、エラトステネスのふるい (sieve)、8-Queen(8Q)、名前登録 (2tree)、quick sort(qsort) の 5 つである。

rj406 は、基本的に 1 命令 1 クロックサイクルで実行するが、1 命令の平均実行時間を 1 サイクルより長くする原因として (1) 記憶装置の平均アクセス時間が 1 より長くなること、(2) ロード/ストア命令では (1) とは別に必ず 1 サイクル伸びること、があげられる。(1) に関してはキャッシュなどの記憶装置の構成によるので (2) の場合についてのみ考える。ロードあるいはストアが実行される割合を p% とすると、平均実行時間は p% 伸びる。これを表 3のデータに適用すると平均 0.16 ～ 0.42 サイクル伸びる。これを解消するためにはバスを時分割して用いるか 2 組のバスを設ける必要があるが、現在は簡単な構造にするために行なっていない。

表 3: 命令頻度

	hanoi	sieve	8Q	2tree	qsort
load	11.6	8.1	35.1	17.5	14.7
store	13.1	8.1	7.3	10.6	9.2
arith	32.1	26.0	27.0	32.9	41.3
logic	0.0	24.4	1.6	0.2	0.0
shift	23.1	24.5	7.9	0.0	3.2
set	0.0	0.2	0.0	0.0	0.0
alu	55.2	75.1	36.5	33.1	44.5
branch	2.9	8.7	9.0	14.3	10.8
jump	5.8	0.0	1.2	10.7	5.2
b & j	8.7	8.7	10.2	25.0	16.0
no-op(ld)	0.0	0.0	5.4	3.4	0.0
no-op(b&j)	10.2	0.0	5.4	10.3	15.8
no-op	10.2	0.0	10.8	13.7	15.8

rj406 においては、2 つのレジスタあるいはレジスタとイミディエイトとの比較による分岐をハードウェアの複雑さと遅れを考えて実装していない。これらの命令は他の 2 命令で代用できる。そのために 1 サイクル増加し、平均実行時間が伸びる。上記のプログラムをこれらの分岐命令があるものとしてシミュレーションをすると、その出現割合は最も多いもので約 10%であった。したがって、平均 0.1 サイクル伸びるが、全体に対する影響は小さいと考えられる。

謝辞

御指導いただいた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