

# マイクロプロセッサ rj406

6L-5

清藤 麻子 菅原 浩二 青柳 圭祐 大北 大  
 大林 雄次 神成 まき 平田 祐介 安田 浩哉  
 (電気通信大学 情報工学科)

## 1 rj406のアーキテクチャ

rj406は32ビットのload/storeアーキテクチャのRISCであり、5ステージのパイプラインにより基本的に1命令1サイクルの処理速度を実現することが可能である。

命令	タイプ	オペレーション
LB (Load Byte)	i	$rf(rd) := mb[rf(rs) + imm]$
LW (Load Word)	i	$rf(rd) := mw[rf(rs) + imm]$
SB (Store Byte)	i	$mb[rf(rs) + imm] := rf(rd)$
SW (Store Word)	i	$mw[rf(rs) + imm] := rf(rd)$
ADD (ADD)	r	$rf(rd) := rf(rs1) + rf(rs2)$
ADDI (ADD Immediate)	i	$rf(rd) := rf(rs) + imm$
SUB (SUBtract)	r	$rf(rd) := rf(rs1) - rf(rs2)$
SUBI (SUBtract Immediate)	i	$rf(rd) := rf(rs) - imm$
AND (AND)	r	$rf(rd) := rf(rs1) \& rf(rs2)$
ANDI (AND Immediate)	i	$rf(rd) := rf(rs) \& imm$
OR (OR)	r	$rf(rd) := rf(rs1)   rf(rs2)$
ORI (OR Immediate)	i	$rf(rd) := rf(rs)   imm$
XOR (eXclusive OR)	r	$rf(rd) := rf(rs1) \oplus rf(rs2)$
XORI (eXclusive OR Immediate)	i	$rf(rd) := rf(rs) \oplus imm$
SLL (Shift Left Logical)	r	$rf(rd) := rf(rs1) \ll rf(rs2)$
SLLI (Shift Left Logical Immediate)	r	$rf(rd) := rf(rs1) \ll shiftcount$
SRL (Shift Right Logic)	r	$rf(rd) := rf(rs1) \gg rf(rs2)$ (上位ビットに0)
SRLI (Shift Right Logic Immediate)	r	$rf(rd) := rf(rs1) \gg shiftcount$ (上位ビットに0)
SRA (Shift Right Arithmetic)	r	$rf(rd) := rf(rs1) \gg rf(rs2)$ (上位ビットに符号)
SRAI (Shift Right Arithmetic Immediate)	r	$rf(rd) := rf(rs1) \gg shiftcount$ (上位ビットに符号)
BEZ (Branch on Equal to Zero)	i	if $rf(rs) = 0$ then $pc := pc + imm + 4$
BNZ (Branch on Not equal to Zero)	i	if $rf(rs) \neq 0$ then $pc := pc + imm + 4$
BPL (Branch on Plus)	i	if $rf(rs) > 0$ then $pc := pc + imm + 4$
BMI (Branch on Minus)	i	if $rf(rs) < 0$ then $pc := pc + imm + 4$
J (JUMP)	j	$pc := pc + offset + 4$
JR (Jump to Register)	r	$pc := rf(rs)$
JAL (Jump And Link)	j	$rf(30) := pc + 4; pc := pc + offset + 4$
JALR (Jump And Link Register)	r	$rf(30) := pc + 4; pc := rf(rs)$
TRAP (TRAP)	r	$rf(rd) := pc + 4; pc := 8$
RETI (RETurn Interrupt)	r	$pc := rf(rs)$

命令タイプ

i-type op(31..26) @ rs(25..21) @ rd(20..16) @ imm(15..0)

r-type op(31..26) @ rs1(25..21) @ rs2(20..16) @ rd(15..11) @ func(10..6) @ shiftcount(5..0)

j-type op(31..26) @ offset(25..0)

表1 命令セット

算術論理演算命令は加・減演算、論理演算及び算術・論理シフト演算がレジスタ間と16ビットイミディエイト値で実行できる。オーバーフローのチェックは行っていない。乗・除演算、浮動小数点演算は演算実行に複数サイクルを必要とするため外部コプロセッサで処理を行わせることにし、内部では処理を行わない。そのためにコプロセッサに対する命令が用意されている。

ロード・ストア命令はデータサイズとしてワード(32ビット)とバイト(8ビット)の2種がある。ロード・ストア命令はバスを占有するため後の命令のフェッチが行えない。このため最低1クロックのストールをする。また、ロード命令はディレイドロード(delayed load)であり、直後の1命令はこのロード命令で読み出された値を使用することが出来ない。

ブランチ命令とジャンプ命令は遅延方式(delayed branch)であり直後の1命令は分岐するしないに関わらず実行される。遅延方式にすることで1命令1サイクル実行が可能となるがこのためにPCブロックの回路が複雑になっている。ブランチ命令はレジスタの内容の零非零・正負の条件判定を行う。

Microprocessor rj406

Asako SEITO, Kouji SUGAWARA, Keisuke AOYAGI, Hiroshi OOKITA,  
 Yuuji OOBAYASHI, Maki KANNARI, Yuusuke HIRATA, Kouya YASUDA  
 University of electro communications

rj406ではLHI (load high immediate) 命令とオーバーフロー検出をする演算命令は実装されていない。

## 2 論理回路

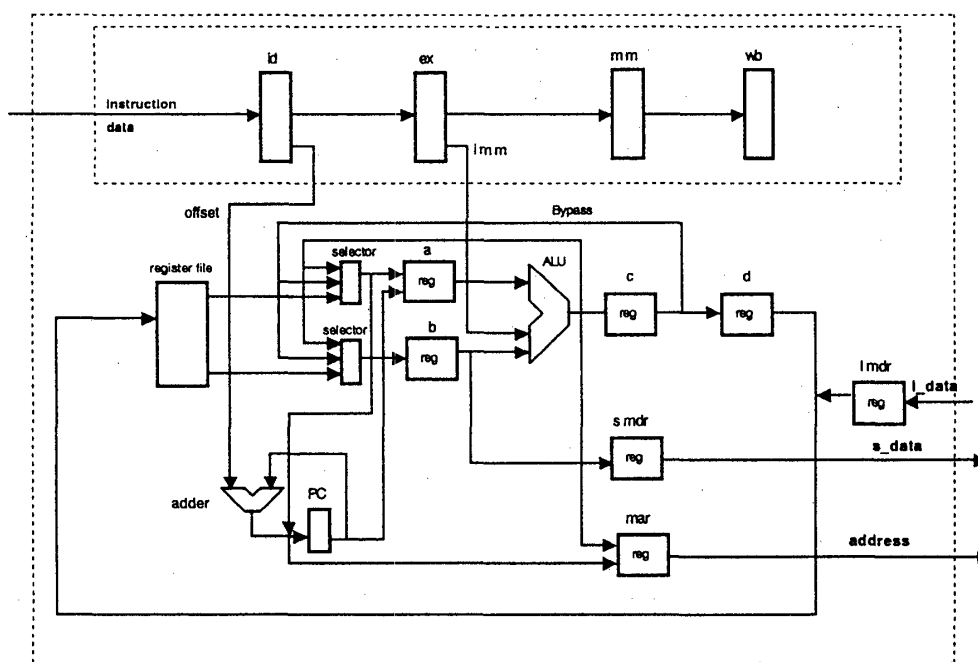


図1 データバス

ES2 (European Silicon Structures) 社のCADツールSOLO1400を用いて設計を行った。

レジスタファイルは1クロック周期で書き込みと読み出しが行えるRAMを使って作られた。このため、IF (命令フェッチ)、ID (命令デコードおよびレジスタからのデータの読み出し)、EX (演算)、MM (メモリロード・ストア)、WB (レジスタへのデータの書き込み) の5つのステージが必要になった。32個あるレジスタのうち0番は値0に固定されている。また30番はリンク付ジャンプ命令に、31番は割り込みで使用される。PCブロックには加算器が含まれており、これでPCのインクリメントと飛び先の計算を行う。

線幅1.2ミクロンルールで演算部分は6735ゲートで37.36 mm<sup>2</sup>、制御部分は命令レジスタを含んで859ゲートで2.04 mm<sup>2</sup>である。全体をまとめた大きさは38.15 mm<sup>2</sup>であるが、面積が小さくなっているのは入出力信号の数のためと思われる。入出力パッドをつけたものは57.38 mm<sup>2</sup>となった。

今後はキャッシュを含めたメモリ関係や乗除演算、浮動小数点演算を処理するコプロセッサなどの設計を行う予定である。

rj406の設計は修士課程と学部の学生計8人のプロジェクトで約3ヶ月を要した。ご指導いただいた中川圭介助教授に感謝致します。また、株式会社物産システムテクノロジーにはLSIの実現に際してご協力をいただきました。ここに謝辞を表します。