システムソフトウェアと
オペレーティング・システム 69−14
（１９９５．６．２３）

超並列計算機 RWC-1 の命令セットアーキテクチャ

岡本 一晃　松岡 浩司　廣野 英雄　横田 隆史　坂井 修一

技術研究組合 新情報処理開発機構 つくば研究センタ

我々は、通信と演算を融合することで局所演算性能と並列処理性能の両者を最適化する、並列処理向けのプロセッサアーキテクチャ RICA(Reduced Interprocessor-Communication Architecture) を提案しており、これに基づいた超並列計算機 RWC-1 の開発を進めている。 RWC-1 の要素プロセッサは、スーパスカラ実行による局所演算処理の高速化を実現すると同時に、ハードウェアサポートによる通信・同期オーバヘッドの削減によって、大域的な並列処理性能の向上を図っている。さらに I/O を高速化し、かつ I/O 処理によってプロセッサ間通信が阻害されないよう、独立の I/O インタフェースを備えている。本稿では RWC-1 プロセッサの命令セットアーキテクチャについて述べ、局所演算および大域並列処理の基本動作を示す。

Instruction Set Architecture of a Massively Parallel Computer RWC-1

Kazuaki OKAMOTO　Hiroshi MATSUOKA　Hideo HIRONO　Takashi YOKOTA
Shuichi SAKAI

Tsukuba Research Center, Real World Computing Partnership
Tsukuba Mitsui Building 16F, 1-6-1 Takezono,
Tsukuba-shi, Ibaraki 305 Japan

We have proposed a high-performance processor architecture RICA(Reduced Interprocessor-Communication Architecture), which decreases communication overhead by fusing thread execution and communication. Massively parallel Computer RWC-1 which consists of 1,024 PEs is now under construction. RWC-1 processor based on the RICA optimizes a local execution within a thread by super-scalar, and also optimizes a global execution among threads by decreasing communication and synchronization overheads with low cost hardware. This paper describes the instruction set architecture of the RWC-1 processor. It also presents how local execution and global execution are optimized.

1 はじめに

我々は新情報処理開発機構(RWCP)において、中長期的な展望に立った汎用超並列計算機の研究・開発を行っており、その第一段階として、要素プロセッサ数1000規模の超並列計算機RWC-1の開発を進めている[1]。

一般に計算機のアーキテクチャを構築する際、その高性能化へのアプローチには2通りの方向が考えられる。一つは命令セットの単純化やスーパスカラ化などによってプロセッサ単体の性能を向上させる方向であり、もう一つは並列化によって台数効果を引き出す方向である。この両者は互いに直交する方式であるため、両者をともに生かすようなアプローチが高性能なプロセッサアーキテクチャを構築する上で必要であると考えられる。我々はすでに両者を最適に満足する独自のプロセッサアーキテクチャRICA(Reduced Interprocessor-Communication Architecture)を提案している[2]。

本稿では、このRICAに基づいたRWC-1用要素プロセッサの命令セットアーキテクチャについて述べる。

2 超並列計算機の要素プロセッサ

超並列計算機の要素プロセッサに求められる要件は、高速性と高機能性である。このうち高速性は、1) 局所実行の高速化、2) 通信インタフェースの高速化、3) 同期の高速化、4)I/Oインタフェースの高速化、5) 負荷分散や並列性制御などの効率的支援、などにより達成される。我々は上述の1)2)3)を融合するアーキテクチャとしてRICAを提唱し[2]、また5)としてはSuper-Threadingを提唱している[3]。一方、要素プロセッサの高機能性には、1) 並列処理プリミティブを自然に実現する命令セット、2) 大域的仮想化、3) 優先度制御の支援、4) 時分割・空間分割の支援、5) 実時間実行の支援などが必要であり、RWC-1ではこれらの諸要素が実現される[1]。

これらの要件を満たすRICAアーキテクチャは、(a) オーバヘッド0のメッセージ処理・スレッド起動、(b) ハードウェアによる高速マイクロ同期機構、(c) メッセージ生成命令と専用パイプライン、(d) スーパスカラ機構、(e) 高速文脈転換機構、(f)(a)～(e)の融合と単純化、から成っている[2][4]。

3 RWC-1の命令セットアーキテクチャ

3.1 要素プロセッサの基本構成と基本動作

RWC-1における1つの処理ノードは、1) 演算処理を行う要素プロセッサ、2) 他ノードとの通信メッセージのルーティングを司るスイッチングユニット、および3) 外付けメモリから成る。このうち要素プロセッサは図1に示す通り、5つの機能ブロックと2つの内蔵キャッシュメモリで構成される。

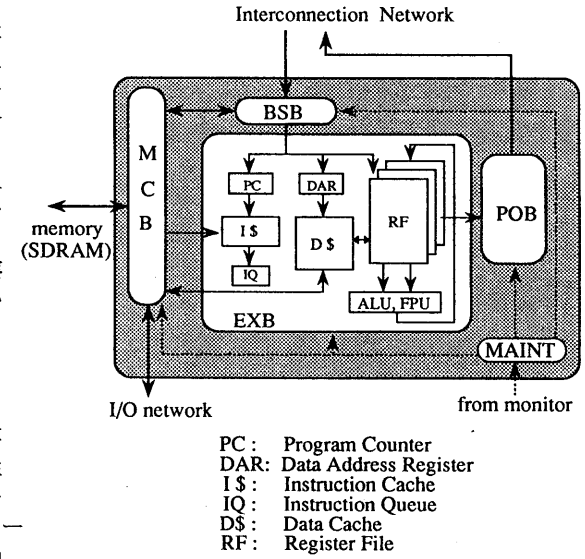

図1: RWC-1プロセッサの機能ブロック

それぞれの機能ブロックは、以下のような機能を持つ。

- BSB (Buffering and Scheduling Block)
  パケットの到着によるスレッドの起動を制御するブロック。
- EXB (EXecution Block)
  演算実行を行うブロック。64ビットRISCアーキテクチャをとり、ALU、FPU、整数乗算回路、整数除算回路、整数除算回路、バレルシフタ、TLB、命令キャッシュ、およびデータキャッシュを有す。また32ワードの汎用レジスタを3セット持っている。これらがスーパスカラ動作する。
- MCB (Memory Control Block)
  外部メモリとの入出力を司るブロック。命令キャッシュやデータキャッシュの入れ換えの

他、パケットの退避や外部I/Oからのメモリアクセスなどを制御する。I/O自体の高速化を図ると同時にI/O処理にプロセッサ間通信を阻害されないよう、結合網とは独立にI/Oポートを用意している。

- POB (Packet Output Block)
  パケットの生成、およびその出力制御を行うブロック。演算実行部とは独立して動作するので、処理の重畳化が可能である。
- MAINT (MAINTenance block)
  チップの保守を行うブロック。実動作とは直接関係ないが、チップの動作を独立にモニタしてテストやデバッグを支援し、さらに必要に応じてボトルネック検出や効率測定などを行う。

RWC-1の要素プロセッサは、スレッドと呼ばれる一固まりの命令列を基本単位として動作し、その動作形態は大きく2つの階層に分けられる。上位の階層はRICAアーキテクチャに基づく通信と演算の融合動作であり、複数のスレッドを多重化して実行する。一方下位の階層は、スーパースカラ実行に基づくスレッド内の局所演算処理である。

RICAアーキテクチャでは、パケットの到着によってオーバヘッドなく自動的にスレッドが起動される。ひとたびスレッドが起動されると、スレッド内の命令列はスーパスカラ処理され、必要に応じて新たなパケットの生成・送出を行う。一つのスレッドは、break命令を実行することで終了し、生成されたパケットは行先で新しいスレッドを起動する。

またRWC-1では、スレッドの起動に際し優先度を設けている。ユーザおよびシステムの各モードで2レベルずつ計4レベルの優先度があり、現在実行中のスレッドの優先度より高い優先度のパケットが到着した時は、実行中のスレッドはプリエンプトされ、新たなスレッドが起動される。ただしシステムモードの2レベル間ではプリエンプションは起こさない。そして、スレッド切替のオーバヘッドを削減するために、RWC-1では3セットのレジスタファイルを用意している。

3.2 命令形式とパケット形式

RWC-1プロセッサの命令は全部で53種類である。命令の一覧を表1に示す。

整数演算、論理演算、ビット演算、浮動小数点演算、メモリ参照、および分岐命令はスレッド内の局所実行で用いられる命令であり、一般的なRISCプロセッサと同様である。これに対し、並列処理用に通信・同期の命令があって、本プロセッサの特長を示すものとなっている。さらに、外付けメモリとI/Oデバイス間でのバースト転送をサポートするI/O命令を実装している。

表1: RWC-1命令一覧

種別	命令			
整数演算	add	sub	mul	divi
	div8	divs	divq	divr
	cmp	seth		
論理演算	and	or	xor	sll
	srl	sra		
ビット演算	ext	ins	ff1	
浮動小数点演算	fadd	fsub	fmul	fdiv
	fcmp	f2i	i2f	
分岐	bra	bcnd	bbs	jmp
	jmpr	jmpc	break	halt
	trap	rte		
メモリ参照	ld	st	lock	
通信・同期	mkpkt	mkpkti	mkcnt	sia
	bsync			
I/O	bgnio	stpio	mkmsg	
その他	mgr	mrg	sgr	prgm
	prgp	cwb		

RWC-1の命令形式は、図2に示す通り大きく5つに分類される。

各フィールドの内容は次のとおりである。

- op(ope-code):inst(31:26)
  演算コード
- w(extended immediate flag):inst(25)
  この命令がxim(拡張即値)を採る/採らないを指定する。
- rd(destination register):inst(24:20)
  演算結果が格納されるレジスタ番号
- rs1(source register 1):inst(19:15)
  第一オペランドのレジスタ番号
- rs2(source register 2):inst(13:9)
  第二オペランドのレジスタ番号
- bcc(branch condition and control bits):inst(24:20)
  分岐条件と分岐制御の指定。
- jc(jump control bits):inst(24:23)
  ジャンプ制御の指定。
- pktc(packet control bits):inst(24:20)
  パケットの優先度とwait conditionの指定。
- op2(auxiliary ope-code):inst(8:0)
  補助演算コード
- eim(embedded immediate):inst(13:0)
  即値

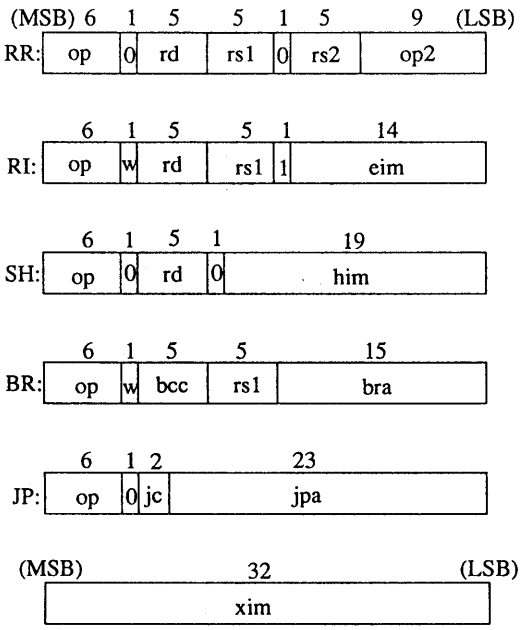

図 2: RWC-1 の命令フォーマット

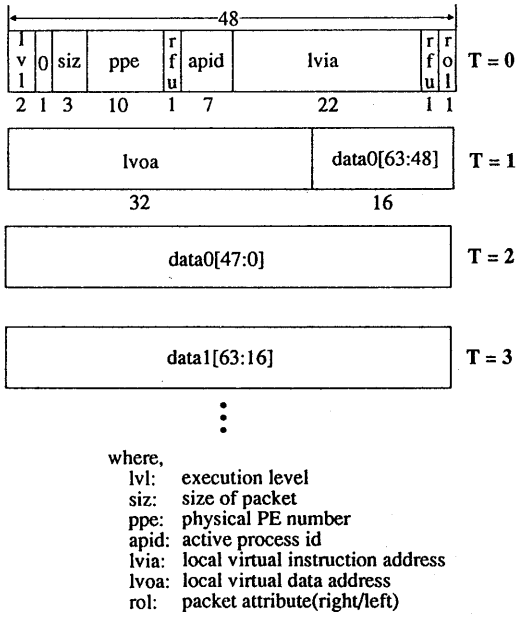

図 3: RWC-1 のパケット形式

- him(immediate high):inst(18:0)
- bra(branch address):inst(14:0)
  分岐先の相対アドレス
- jpa(jump address):inst(22:0)
  ジャンプ先の絶対アドレス
- xim(extended immediate):xim(31:0)
  拡張即値

RWC-1 では、プロセッサ間の通信をパケットにより実現している。パケットは行き先のプロセッサ番号、命令アドレス、データアドレス、優先度などから成るヘッダ部と、1～8 ワードの間で可変長のデータ部で構成される (図 3)。RWC-1 は 64 ビット計算機であるが、パケットの語長は 48 ビットとなっている。結合網はパケット 1 語を 1CPU クロックで転送する。

3.3 局所演算処理に関する命令

スレッド内の局所実行は、スーパスカラアーキテクチャに基づいて行われる。一般的なプロセッサと同様に、整数演算命令、論理演算命令などを備え、また 64 ビットの浮動小数点演算命令も実装する。分岐命令には遅延分岐は採用せず、かわりに hint bit を設けることで最適化する。hint bit の真偽により、先行フェッチする命令アドレスを決定する。

それぞれの演算処理部 (パケット生成を含む) は独立のパイプラインを持っており、特にデータ依存関係がない限り並行して動作することが可能である。RWC-1 では、2 命令同時発行によるスーパースカラ実行を実現している。各命令におけるパイプライン起動時間は 1 クロックである。

3.4 大域並列処理に関する命令

RWC-1 では、プロセッサ間の通信に関わる並列処理命令として、1) コンティニュエーションの生成、2) パケットの生成、3) 同期、の 3 種類の命令を用意している。

mkcnt 命令はコンティニュエーションを生成する命令である。コンティニュエーションとは新たなスレッドを起動する際に必要となる、命令アドレスとデータアドレスとの組合せである。本命令で生成されるコンティニュエーションは、一度汎用レジスタに格納される。コンティニュエーションは、後述の mkpkt 命令でパケットに埋め込まれて転送される。

mkpkt 命令は通信のためのパケットを生成する命令である。上述のコンティニュエーションと、1～8 ワードの任意の汎用レジスタの内容をパケット化して転送する。この際、コンティニュエーションの論理アドレスを GTLB(Global TLB) でアドレス変換して、行き先の物理 PE 番号を得る。本

命令で起動されるパケットの生成・送出パイプラインは、命令の実行とは独立に動作するので、RISC の一命令として起動された後はデータの依存関係がない限り後続の命令実行と並行して動作する。パケット生成・送出パイプラインでは、

1) コンティニュエーションから GTLB を用いて送り先の物理 PE 番号を獲得し、ヘッダ部を生成する。
2) register file から転送データを読み出し、データ部とする。
3) 64bit → 48bit 変換をする。

が行われる (図 4)。

mkpkt 命令はすでに生成されているコンティニュエーションからパケットを生成するが、mkpkti 命令は自身でコンティニュエーションを生成する複合命令である。通常 mkpkti 命令が生成するコンティニュエーションは、パケットの行き先での新規スレッドの起動に用いられ、mkcnt 命令で生成されるコンティニュエーションは、主に関数コールなどの際にリターンアドレスとして相手側に送られる。

同期命令は複数のスレッドやプロセス間で同期を取る際に必要となる。RWC-1 では同期処理に対しても RICA 的な考え方を採り、最も単純な 2 パケット間の同期を取る bsync 命令だけを実装している。そしてこの binary synchronization を組み合わせることにより、さまざまな同期形態に柔軟に対応していく。bsync 命令は、先着したパケットを一度メモリ上に退避し、後続のパケットが到着したら退避したパケットを再びレジスタ上に戻して同期を成立させる命令である。データフロー計算機における direct matching 機構を命令で実行するものである。

3.5 I/O 命令

RWC-1 ではプロセッサ間の通信が I/O の入出力処理によって阻害されないように、I/O 用の結合網を独立させている。RWC-1 の I/O 転送は、基本的にはメモリ・メモリ間もしくはメモリ・デバイス間でのバースト転送であり、プロセッサ内の必要な制御レジスタを設定した後に bgnio(begin I/O) 命令、stpio(stop I/O) 命令を実行することで、バースト転送を開始、もしくは終了させる。この他 I/O デバイス側の制御レジスタを設定する手段として、mkmsg(make message) 命令を用意している。これらの I/O 命令も、シーケンサは独立した I/O 処理パイプラインを 1 クロックで起動するだけであり、その後の I/O 処理は後続の命令実行と並行して行われる。

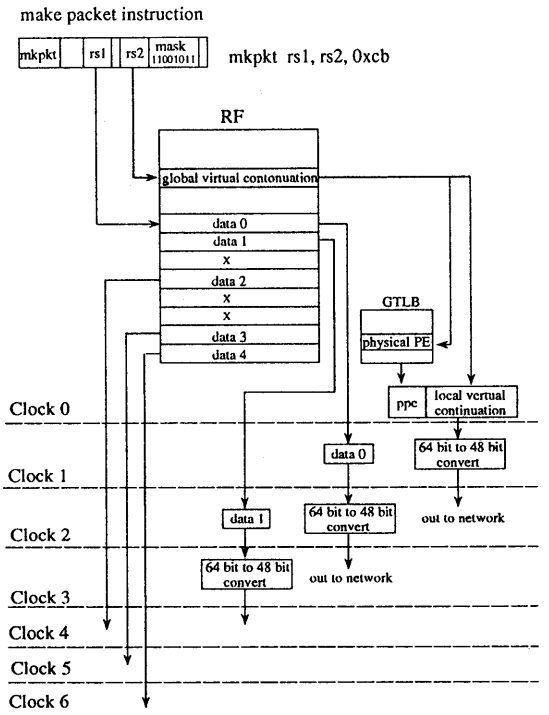

図 4: mkpkt 命令と動作

3.6 プログラムの実行例

RWC-1 の動作例を示すため、2 つのリモートデータをパケットにより受けとり、それを処理してリモートメモリに書き込むプログラムを考える (図 5)。

受けとった 2 つのパケットデータは、bsync 命令により同期がとられる。この時 bsync 命令は、2 つのデータを汎用レジスタの r0 と r8 に注入する。またレジスタ r31 はゼロレジスタである。同期がとれた後の処理パイプラインの動作を、図 6 に示す。ここでは自身のメモリへの書き込みが、キャッシュにヒットした場合を示しているが、ミスヒット時の入れ換え操作もすべてハードウェアがサポートしているので、後続の add 命令、mkcnt 命令、および mkpkt 命令の実行が妨げられることはない。

4 ハードウェア実装

本稿で述べた命令セットアーキテクチャに基づく RWC-1 用の最初のプロセッサチップを、0.6μm ルールの CMOS スタンダードセル上に実現した。ゲート規模は、ランダムロジック部分が約 11 万

処理側:

```
        bsync   $r30
        fadd    $r1, $r0, $r8
        add     $r2, $r31, write_adr0
        fsub    $r3, $r0, $r8
        mkcnt   $r5, remote_cnt0
        mkpkt   $r1, $r5, 0x03
        st      $r2, $r1
        add     $r4, $r31, write_adr1
        mkcnt   $r6, remote_cnt1
        mkpkt   $r3, $r6, 0x03
        st      $r4, $r3
```

リモート側:

```
remote: st      $r1, $r0
        break
```

図 5: プログラム例

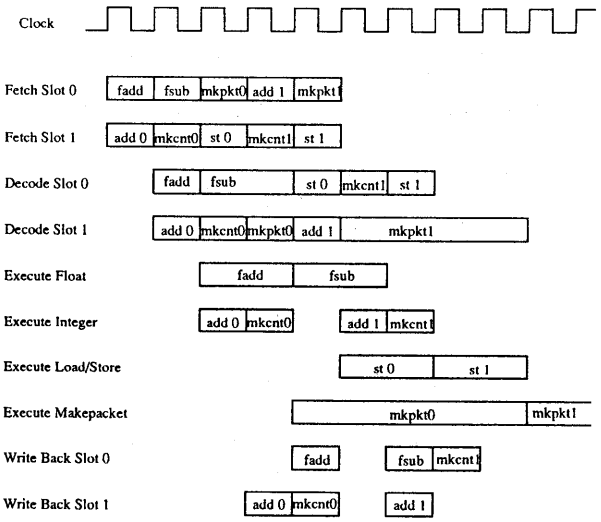

図 6: 処理パイプラインの動作

ゲート、他に内蔵キャッシュメモリを 2KB 実装している。パッケージには 447PIN の PGA パッケージを採用し、121 信号をルーティング用のスイッチチップとのインタフェースに、92 信号を外部メモリ (Synchronous DRAM を採用) とのインタフェースに、19 信号を I/O インタフェースに、20 信号をシステム制御用に、残りを電源およびグランドに割り当てている。またプロセッサの内部は動作周波数 50MHz で設計されている。

なお、上述の通り最初のプロセッサはすでに開発済みであるが、ハードウェア量の制約から以下の機能を削減している。

- 浮動小数点演算命令
- 同期命令
- 整数除算命令
- 複数命令同時発行機構

これらの機能については、現在並行して開発を進めている次のプロセッサチップ (ランダムロジック 20 万ゲート /527 ピン) でフルに実装中である。

5 おわりに

本稿では、超並列計算機 RWC-1 の命令セットアーキテクチャについて述べた。RICA に基づくプロセッサアーキテクチャをとることにより、局所演算性能と大域並列処理性能の両方の高速化を図っている。さらに高速なマイクロ同期機構や独立の I/O 機構を採用することで、種々の並列処理演算に柔軟に対応している。

本プロセッサの第一バージョンはすでに開発済みであり、現在最初のテストベッドに実装して検証中である。並行して開発を進めている次のプロセッサチップには、ハードウェア量の制約から第一バージョンで削減されていた機能を実装するとともに、現在行っている検証結果を反映させていく予定である。

謝辞

本研究を遂行するにあたり、有益な御指導、御討論をいただいた島田つくば研究所長、石川超並列ソフトウェア研究室長、超並列ソフトウェア研究室員の諸氏、ならびに RWC 超並列アーキテクチャワーキンググループの諸氏に感謝いたします。

参考文献

[1] S.Sakai, et.al., RWC-1 Massively Parallel Architecture, Proc. HPCC'94, pp.33-38 (1994).

[2] S.Sakai, et.al., Reduced interprocessor- communication architecture and its implementation on EM-4, Parallel Computing Vol.21, No.5, pp.753-769 (1995).

[3] S. Sakai, et.al., Super-Threading: Architectural and Software Mechanisms for Optimizing Parallel Computation, Proc. ICS'93, pp.251-260 (1993).

[4] 松岡他、超並列計算機 RWC-1 用プロセッサチップの設計、信学技報 CPSY95-18 (1995).