

超並列計算機 RWC-1 における入出力機構

廣野 英雄† 松岡 浩司† 岡本 一晃† 横田 隆史*
堀 敦史† 児玉 祐悦† 佐藤 三久† 坂井 修一†

†(技組) 新情報処理開発機構 つくば研究センタ

* (技組) 新情報処理開発機構 超並列三菱研究室

†電子技術総合研究所

概要 汎用超並列計算機 RWC-1 における入出力機構は高速かつ大量データの扱いが可能だけでなく、時分割・空間分割、リアルタイム処理などに対応していることが必要である。この要件を満たすために入出力機構には、命令実行系と独立した階層構造を持つ相互結合網が有利であることが分かった。本稿では下位階層の相互結合網として優先度付きマルチチャンネルリングバスを提案する。本相互結合網は通常のリングバスに優先度処理を加えることで高優先度の入出力パケットの転送時間を保証し、またリングを分散・集中混在型の方法で効率良く制御することが可能となった。

I/O System for the Massively Parallel Computer RWC-1

Hideo HIRONO† Hiroshi MATSUOKA† Kazuaki OKAMOTO† Takashi YOKOTA*
Atsushi HORI† Yuetsu KODAMA† Mitsuhsa SATO† Shuichi SAKAI†

†Tsukuba Research Center, Real World Computing Partnership

Tsukuba Mitsui Building 16F, 1-6-1 Takezono,

Tsukuba-shi, Ibaraki 305, Japan

* Massively Parallel Systems Mitsubishi Laboratory, Real World Computing Partnership

†Electrotechnical Laboratory

Abstract The RWC-1 should perform efficient I/Os under time/space sharing and real time environments, as well as it should realize fast and high-throughput data transfer. To satisfy these requirements, RWC-1 I/Os had better have a hierarchical network which is independent of the data transfer in the upper memory hierarchy. This paper proposes a multiple-channel ring bus for a lower level I/O network. The priority handling mechanisms are attached to the ring bus which can keep the maximum transfer time for high-priority I/O packets. The ring bus is controlled in a distributed / centralized way which brings excellent I/O performance.

1 はじめに

筆者らはリアルワールドコンピューティング (RWC) 研究計画のシステム基盤として、汎用的な超並列計算機の検討・開発を行なっている。その最初のプロトタイプとして1996年3月までに1000台規模の要素プロセッサ (PE) からなる超並列計算機 RWC-1 を開発する [1]。これは一般の計算機としての能力に加えて実時間性、スタンドアロン性、柔軟性を重視し、RWCの中心的課題である。「柔らかな情報処理」が効率良く実行できる環境を目指している。

一般に計算機の性能といえば数値演算能力を注目しがちである。しかし実際のアプリケーションを実行した際に現れる性能は、それだけではなく入出力機構を含めたシステム全体の性能に依存する。これは、いくら数値演算能力が高い計算機であっても、システムのバランスが悪ければその能力を生かすことが出来ないことを意味する。

超並列計算機の場合、一般に膨大な量のデータを扱うアプリケーションが多くデータの物理的配置も分散しているため、一般の計算機以上に入出力機構に比重がかかっているといえる。したがって RWC-1 の入出力機能は全体性能の鍵を握るものであり、アプリケーション、演算性能とのバランスの面から、十分検討されなければならない。

本稿では「柔らかな情報処理」にも対応が可能な超並列計算機 RWC-1 の入出力機構を提案する。2節では超並列計算機に必要な機能と性能を考え、3節では RWC-1 に適した入出力機構とはどのようなものかを検討し、4節では本稿で提案する優先度付きマルチチャネルリングバスについて説明する。

2 超並列計算機 RWC-1 の入出力機構の要件

2.1 RWC-1 の特徴

RWC-1 の特徴を以下の4つの点から述べる。

1. アーキテクチャ

RWC-1 は計算と通信を融合した新しいアーキテクチャ Reduced Interprocessor-Communication Architecture (RICA) に基づいた計算機である。RICA では、通信・メッセージ処理に関するオーバーヘッドを軽減させるため、これらを専用ハードウェア化し、命令実行と非同期パイプライン化している。

RICA においては PE に到着したパケットは必要に応じて同期が取られるが、同期は専用ハードウェア

により高速に行なわれる。その後パケット中のデータは自動的にレジスタにロードされる。パケットの生成も専用ハードウェアによって演算と非同期に行なわれる。

RWC-1 では、このように通信コストが小さい RICA の採用によって効率の良い細粒度並列処理を実現することが可能になっている。

2. ハードウェア

RWC-1 は1000台規模の PE からなる MIMD 型超並列計算機である。集中型の共有メモリはなく、高速のメッセージ処理を介して分散共有メモリモデル、メッセージ交換パッシングモデル、データ並列モデルなど様々な計算モデルを支援する。

RWC-1 を構成する PE は自主開発の 64bit RISC プロセッサであり、1台あたり 32~64MB のメモリと高速キャッシュを持ち、約 100MFLOPS (倍精度) の演算能力を持つ。さらに MMU を内蔵し、大域的な仮想記憶をサポートしている [2]。

またそれぞれの PE は内部に演算系と独立動作する通信機構を持ち、低レイテンシ、高スループットの CCCB (Cube-Connected Circular-Banyan) 結合網で結ばれている [3]。

3. ソフトウェア

RWC-1 は自分自身が超並列 OS を持つスタンドアロン・システムである。RWC-1 は大域的仮想化、マルチユーザ・マルチタスク、空間・時間分割、優先度処理など超並列 OS に必要とされる様々な機能を有している。

プログラミング言語としては、システム記述用に C++ の拡張版を、高位言語としてオブジェクト指向言語などを考えている。

4. アプリケーション

RWC-1 は RWCP において研究されているパターンと記号の融合処理、具体的には、画像理解・音声認識・ロボットなどの分野の様々なアプリケーションのためのプロトタイプ計算機と位置付けられている。このため、数値計算用の超並列計算機に必要なかった不定型の処理、実時間処理などが行われる。

また、従来の超並列計算機のアプリケーションであった流体、QCD などの数値計算も高速に実行されなくてはならない。つまり、RWC-1 は、以上の意味の汎用超並列計算機であるといえる。

2.2 RWC-1 の入出力の要件

RWC-1 を実現するのに必要な入出力機構の要件を考える。

1. 高いバンド幅

RWC-1 では入出力の高いバンド幅が要求される。一例をあげれば、HDTV の信号の処理では、1 ポートで約 200MB/s のバンド幅が要求される (24bit のフルカラー)。ディスクに関しては、並列化によって高信頼化・高スループット化を図るため、数 10 から 100MB/s 程度のバンド幅が必要となる。その他、音声情報の処理にも高いバンド幅が要求される場合がある。

2. 最適な分散配置とアクセス時間のばらつき抑制

RWC-1 では複数の PE で様々な処理が行なわれているため、入出力の要求も同時に多数の場所で起こる。これに対応するために入出力を分散配置し、これらの要求を効率良く並列に処理しなければならない。そのためには、入出力装置の物理的分散ばかりでなく、ソフトウェアによる入出力装置の割り付けの最適化が必要となる。

一方で、スケジューリングの限界などから、PE と入出力装置の位置関係によって、アクセス時間に著しいばらつきが生じないように構成にする必要がある。

3. 実時間性

先に述べたように、RWC の応用では、実時間性が重要となる。このために入出力機構には入出力データの転送時間を保証するなどの機構が必要である。

4. 大域的仮想記憶の支援

RWC-1 では PE 番地を含む大域的仮想記憶を実現する。そのため、高速のページングを支援する機構が入出力に必要である。またコアとディスクを含めた広域一次元アドレスを検討中であるが、この場合、ページングの頻度が増すために、入出力の高速化はさらに重要な課題となる。

5. 時分割・空間分割への対応

RWC-1 では高速の時分割処理を支援している。このため、現在実行中のプロセスを決められた時間内に退避し、新しいプロセスを起動する機構が必要であり、入出力はそのための機構を提供する必要がある。

また RWC-1 では空間分割により複数のプロセスを同時に実行することが可能である。もしそれら

が実時間処理が必要なプロセスであれば、それぞれの分割領域ごとに実時間性を保証しなければならない。これは入出力機構の配置が空間分割に対応し、かつ入出力の処理が他の分割領域に影響を与えてはいけないことを意味する。

6. ディスク容量

目下、RWC-1 では 1 台の PE あたり数 100MB から数 GB、システム全体で数 100GB から数 TB のディスクの容量が必要だと考えられている。

7. 信頼性

一般の計算機では信頼性の向上のためにテープドライブなどによるバックアップの手段がとられている。しかし RWC-1 ではディスク全体の容量が数 100GB から数 TB と膨大であるため、頻繁にディスク全体のバックアップを取ることは難しい。また、ディスクの数が多くなるためシステムで見た場合ディスクの故障する確率は高くなる。このため、ディスク自体の信頼性が高い、MTBF の長いものが必要である。RAID の導入などの他に、分散配置されたディスクへの冗長度のある格納方式などが検討の対象となる。

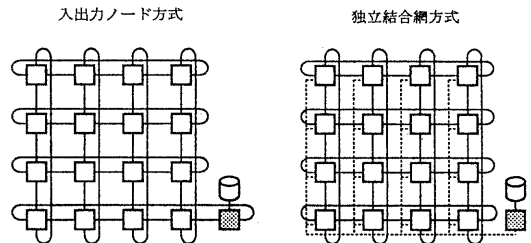


図 1: 入出力ノード方式 vs 独立結合網方式

3 RWC-1 の入出力機構

本節では、前節で述べた要件を満足する入出力機構の構築を検討する。

3.1 入出力ノード方式 vs 独立結合網方式

超並列計算機における入出力機構の実現方式には入出力ノード方式と独立結合網方式に大別される。ここでは RWC-1 の入出力機構としての両者の利害得失を比較検討する。(図 1)

● 入出力ノード方式

入出力ノード方式は入出力機構を命令実行系の相互結合網の中の入出力ノードとして実装する方式

である。商用計算機では、CM-5、Paragonなどがこの方式を採用している [4][5]。

この方式では結合網が単一で済むことから、ハードウェアコストが小さく、PE ノードに対しても入出力ノードに対しても統一的な操作ができるという利点がある。

しかし、(1) 入出力データが結合網を占有して命令実行系のデータの転送が阻害される、(2) 結合網ノードの不均質性から、負荷分散・データ分散・アドレッシングなどが不自然で非効率的になる、(3) 実時間性が要求される入出力処理を命令実行系の相互結合網が行うのは困難である、などの問題点がある。

- 独立結合網方式

独立結合網方式は、命令実行系とは独立の入出力結合網を設け、各 PE が入出力結合網の通信インタフェースをもつ方式である。これには nCUBE3 や Parsytec GC などが当てはまる [6] [7]。

本方式の場合、新たに入出力結合網のためのハードウェアコストが必要になる反面、入出力データは独立した結合網を通じて転送されるため、命令実行系のデータ転送を妨害することがない。また、実行系の結合網を均質なものと構築することができ、負荷分散・データ分散・アドレッシングなどが自然で効率的なものとなる。

RWC-1 では、以下に述べる理由で独立結合網方式を採用ことにした。

1. 転送データの性質の違い

命令実行系の転送データは一般に演算結果や制御のためのデータなどであり、大きさが比較的小さく、プログラムの実行に直接結び付いているため高速性が要求される。

一方、入出力系の転送データは、ディスクのアクセスにみられるように大きな大きさのものが多く、また一般にアクセス時間の大きなものが多いため、通信自体のレイテンシは多少大きくても無視することができる。また入出力系の転送データには命令実行系のそれとは違って実時間性を必要とするものがある。

このように命令実行系と入出力系では明らかに転送データの性質が異なり、また RWC-1 は RICA に基づく細粒度超並列計算機であるため、一般の計算機以上にこの傾向が大きい。したがって、それぞれ別の結合網を設けた方が効率が良いと考えられる。

2. 命令実行系結合網のドレイン処理

マルチユーザ・マルチタスクの環境において時分割によるプロセスの切替を行なう際、命令実行系の結合網を一旦すべて止めて結合網上にあるデータを取り出すことが必要となる。これを結合網のドレイン処理という。もし結合網が 1 系統しかなければ、ドレインの開始、終了などで全体の同期を取るものが原理的に不可能である。独立した入出力結合網があれば、命令実行系の結合網が止まった状態ですべての PE 間の通信が可能であり、ドレインのために全体の同期を取ることができる。

3.2 入出力のネットポロジ

入出力結合網のトポロジーについて検討する。

3.2.1 階層構造の導入

RWC-1 の PE 群は、4PE ないし 8PE を最少の単位としてグループ化される [3]。グループ内では、ページングや大きな局所データの読み書きを高速に行う必要がある。また RWC-1 では空間分割が行なわれる。このとき分割領域ごとに独立して動作し、他の領域を乱さないことが必要となる。

これらの理由により、最小の分割領域を単位として全体のプロセッサを複数のクラスタに分割し、クラスタごとにディスクを実装することが望ましいと考えられる。これによって、(1) 局所データ処理の高速化・並列化、(2) 全体の入出力系のコスト軽減が実現される。

また、RWC-1 ではクラスタ間の入出力装置の共有が行われており、クラスタ間結合網が必要となる。すなわち、RWC-1 の入出力結合網は、クラスタ内とクラスタ間の 2 階層の階層構造を持った結合網が適当である。

3.2.2 クラスタ内入出力結合網

クラスタ内入出力結合網は、まず第一に十分に大きなバンド幅を持つことが必要とされる。しかし、入出力は結合網のレイテンシの大きさが全体性能にあまり関係しないため、直径の比較的大きな結合網でも構わない。

リング型結合網は、物理的にはポイントからポイントの転送であるため転送クロックを上げることができ、またノードあたりのポート数も入出力の 2 ポートで済むため、ハードウェア量が節約される。また、アービトレーションがいらぬために制御が簡単であり、ハードウェアコストも小さい。

よって RWC-1 ではクラスタ内入出力結合網としてリング型結合網が望ましいと考えられる。

図 2 にクラスタ内結合網の一例を示した。クラスタ内結合網は PE ノード、DISK ノード、クラスタ間結

合網ノードをもち、それらがリングバスによってつながれた構造を持つ。PE ノードは、命令実行・コマンドのやりとり・メモリのアクセスなどを行なう。DISK ノードにはインタフェースを介して高スループットの RAID が接続されている。クラスタ間結合網ノードはクラスタ間入出力結合網とつながり、クラスタ外の DISK や PE ノード、画像・音声インタフェースノードなどとのアクセスを行なう。

なおクラスタ間結合網は現在検討中である。

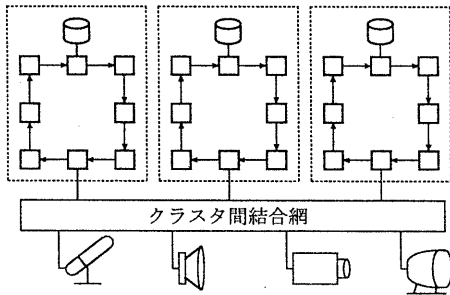


図 2: 入出力結合網の構成

4 優先度付きマルチチャネルリングバスの提案

4.1 優先度付きマルチチャネルリングバスの基本動作

前節で、RWC-1 のクラスタ内の入出力結合網としてリングバスを用いることを述べた。本節では、リングバスの通信方式に関する検討を行う。

4.1.1 パケットとスロット

優先度付きマルチチャネルリングバスでは、固定長のパケットを用いることとした。これは制御を簡単にするためと、DMA(Direct Memory Access) 中心の入出力ではパケットの長さ比べて小さいデータを転送するということが頻繁には起こらないためである。

リングバスは、論理的には円環状のシフトレジスタとみなすことができる。これは、ある一瞬を見た場合、パケットのワード長ごとにグループ分けされている。このグループをスロットという。スロットとパケットは 1 対 1 に対応し、スロットの先頭ノードにはパケットのヘッダがあり、先頭から N 番めのノードにはパケットの N ワード目のデータがある。パケットの内部構成を図 3 に示す

図からわかるように、パケットのヘッダは、宛先を示す ID の領域、パケットの存在を示すプレゼンスフラ

ヘッダ	データ 1	データ 2	データ 3
ID			
プレゼンス			
優先度			

図 3: 入出力パケットの内部構成

グ、優先度を示す領域などをもつ。

4.1.2 パケット転送

パケットの転送は、リングバス上のデータを、1 語分だけ一斉に次のノードに転送することによって行なわれる。この方法ではリング内に FIFO を設ける必要がなく、制御が簡単である。

リングの動作はヘッダの情報に基づいて行なわれる。ヘッダの ID と一致するノードのみがデータをリングから取り出すことができる。他のノードはただ次のノードにパケットを転送するだけである。これによってリングに出力されたデータは、必ず決まった時間で目的ノードに到着することが保証される。

プレゼンスフラグが立っているスロットはパケットを転送中であり、立っていないと転送されていないことを表す。プレゼンスフラグが立っていないスロットを空スロットといい、すべてのノードは空スロットにのみデータを出力できる。

4.1.3 優先度

1 度リングに出力されたデータは目的のノードに到着するまでリングを占有する。このことは、途中にあるノードはパケットがたまっていてもリングに出力できないことを意味する。極端な例では、パケットの転送先が自分の 1 つ手前のノードであれば、空スロットは転送先のノードと出力ノードだけを通り他のノードを通らない。このため、リングにある残りすべてのノードはこの 2 つのノード間の転送が終るまで全く何もできない。入出力系ではディスクからのデータ転送などのようにデータ量が多いため、これらのパケットがリングを占有する不公平は大きな問題となる。

また、入出力パケットにも迅速に処理しなければならないもの、重要なものがある。例えば、実時間処理が必要なパケットは通常のパケットよりも早く処理する必要があるし、コマンドパケットはデータパケットよりも重要である。すべてのパケットには優先度があり、リングはその優先度にしたがって動作しなければならない。

その方法の 1 つは各パケットに優先度を設け、各ノードで流れてきたパケットと待っているパケットの優先度を比較し、もし低ければそのパケットをリングから取り出し、代わりに高い優先度のパケットを出力するもので

ある。しかしこの方法では、取り出したパケットのための余分なバッファがノード内に必要になり、また制御が複雑になる。

ノードから出力できるのは空スロットが来た時のみであるから、高い優先度のパケットが待っているノードに空スロットを与えるような方法を考えればよい。これは、低い優先度のパケットが待っているノードが空スロットが来てもパケットを出力できないように、スロットに優先度を設けることで実現できる。つまりパケットが待っているノードは空きスロットが流れてきた時に、スロットの優先度とパケットの優先度を比較して、同じかパケットの方が高い時のみ出力できる。

空きスロットでなかった場合も、スロットの優先度とパケットの優先度を比較し、パケットの方が高ければスロットの優先度を更新し次回の空きパケットの予約する。この時、更新される前の優先度の値はノード内に記憶しておく。

スロットの優先度は、予約された空スロットにパケットが出力された時に以前記憶していたものと交換されるが、この時ノードに次のパケットが待っていればその優先度と比較して高い方をスロットの優先度とし、次回の空スロットの予約をする。この様子を図4に示す。

aにある高い優先度のパケットはgからdへ転送されるパケットのためにリングに出力できない。そこでaでスロットの優先度をlowからhighに上げる。dから流れた空パケットは優先度highで予約されているため、優先度lowのgはパケットを出力できない。その結果空パケットがaに流れ、aはパケットを出力することが出来る。またaには1つのパケットしかなかったとすると、次回の予約は必要ないのでスロットの優先度はlowに戻る。

このようにスロットに優先度を設けることによって、最も高い優先度を持つパケットは高々1周待つだけでリングにデータを出力することができる。

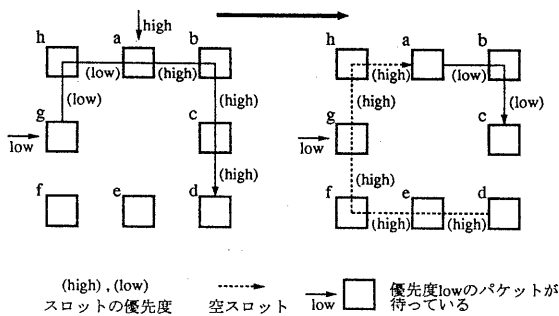


図4: リングバスにおける優先度制御

4.1.4 キャリアパケット

優先度によって優先度の違うノードを選択することはできるが、同一の優先度のパケットを持つノード同士だと選択することができない。また同一優先度のノードは位置関係によって強弱が決まるため、場合によってはある特定のノードのみが出力し続け、全く出力できないノードができることがある(スタベーション)。このため同一優先度のノードであっても、どのノードに出力させるか選択する必要がある。

もし出力させたいノードが自分の直後であれば、単に空スロットを流すことでパケットを出力させることができる。しかしそうでない場合で、自分と出力させたいノードの間に同じ優先度のノードがあった場合、単に空スロットを流しただけでは中間のノードが空スロットにパケットを出力してしまう。したがって、出力させたいノードの直前のノードが空きスロットを流すようにしなければならない。

もし直前のノード宛に何らかのパケットを出力した場合、直前のノードはその自分宛のパケットを取り出すので、直後のノードには空スロットが流れる。このようなダミーのパケットを用いて出力するPEを選ぶことができる。ダミーのパケットが普通のパケットだと直前のノードに何らかの副作用を及ぼすので、何も副作用を起こさないパケットでなくてはならない。このパケットを空スロットを運ぶという意味でキャリアパケットと呼ぶ。

まとめると、優先度付きマルチチャネルリングバスでは直前のノード宛にキャリアパケットを出力することによって、同一の優先度同士でも自由に出力させたいノードを選択することができる。

4.2 リングバスのスケジューリング方式

リングバスのスケジューリング方式には集中管理と分散管理の2種類がある。優先度付きマルチチャネルリングバスでのそれぞれの実現方法を述べる。

1. 集中管理

キャリアパケットを用いることによって予めスケジュールされた通りに出力パケットを選択する集中管理が可能である。図5にこれを示す。

まず、集中管理のために2つの優先度(high, low)を用意する。集中管理に参加するパケットは共通する1つの優先度lowを持つ。

リング中のあるノード(a)がマスタとなる。マスタはスケジュールにしたがって、出力させたいノード(e)の直前のノード(d)宛にキャリアパケットを出力する。この時の優先度をlowとする。これによって、出力させたいノードは空スロットを得

てパケットを出力することができる。もしこれが空スロットであれば途中にあるノード (c) がパケットを出力してしまいうまくいかない。このとき、同時にスロットの優先度を high にしておく。

集中管理に参加するパケットの優先度は low なので、マスタに戻るまで空スロットにはどのノードも出力できない (h)。空スロットを受けとったマスタは、このように繰り返しスケジュールにしたがって出力させたい直前のノード宛にキャリアパケットを出力する。

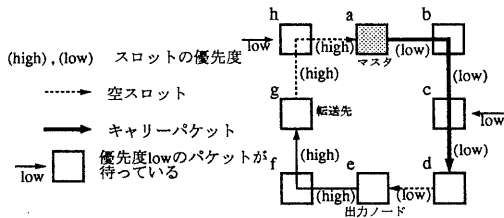


図 5: 集中管理

2. 分散管理

分散管理はそれぞれのノード自身がスタベーションを起こさないように管理をする方式である。

送信中のノードは単位データ数を連続して転送した後、自分の転送のために出力できなかったノードがないかチェックする。その後、出力できなかったノードの数×単位データ数だけ待ち、すべてのノードが出力し終のを待つ。そしてまた単位データ数を出力することを繰り返す。

この方式では集中管理と違って、重ならない限り 1 つのリング上で同時に複数の転送ができる。

分散管理であっても集中管理であっても、途中で実時間処理などの優先度の高いパケットが来た場合、優先度の高いパケットが転送されなければならない。優先度付きマルチチャネルリングバスではいつでも優先度に従った動作をするため、何らハードの追加をすることなしにこれが可能である。

またここで提案した優先度付きマルチチャネルリングバスは同じハードウェアで分散管理と集中管理が可能である。それぞれ目的に応じて使いわけることで効果的な運用をすることが可能である。

4.3 プロセッサの入出力機能と動作

● 入出力機能

RWC-1 での入出力はすべて入出力結合網にパケットを転送することによって行なわれる。これを効

率良く行なうためにプロセッサには様々な入出力の機能が備わっている。

RWC-1 では入出力結合網に 2 種類のパケットを流す。

コマンドパケットはノードに対して様々なリクエストを行なう。プロセッサは入出力命令でコマンドパケットを送信することができ、受信したコマンドパケットによって割り込みが発生する。

また入出力にはディスクのアクセスのように大きなサイズのデータの転送が必要なため、DMA を行なうことが必要である。このため、パケット中にアドレスとデータを持つ DMA パケットがある。プロセッサは DMA パケットを受信すると、自動的にアドレスとして指定されたメモリにデータを書き込む機能や、指定されたメモリの領域からデータを読み込み、自動的にコントロールパケットや DMA パケットを作って送信する機能を持っている。

● 入出力動作

RWC-1 の入出力は基本的には DMA を用いて行なわれる。例えば、ディスクへのライトはプロセッサのメモリ上のデータを DMA によってディスク上に転送することによって行なわれる。

また DMA を用いた入出力ではディスクなどデバイス側に主導権をもたせる方式を検討している。この方式の場合、プロセッサはデバイス側に対してコマンドを発行すると、後はコマンドの終了を待つだけで良い。その間に入出力デバイス側が DMA を用いてデータの転送を行ない、プロセッサは別のタスクを行なうことができる。

5 おわりに

本稿では RWCP で開発中の超並列計算機 RWC-1 の入出力機構について述べた。RWC-1 は RICA に基づく汎用超並列計算機であり、自身で OS を持ち大域的仮想記憶、マルチユーザ・マルチタスク、時間分割・空間分割をサポートしている。またリアルタイム処理を含む様々なアプリケーションに適した最初のプロトタイプとして位置付けられている。

本稿の検討の結果、これらの特徴を生かすためには、命令実行系とは独立したネットワークで複数のクラスタに分割された階層構造を持つことが有利であることがわかった。さらに、クラスタ内ネットワークとして優先度付きマルチチャネルリングバスを提案した。これは優先度処理に対応し、分散・集中管理の混在が可能なリング

型ネットワークであり、両管理方式を適宜使い分けることにより、リアルタイム処理を含むさまざまなアプリケーションの実行に適している。

今後の課題としては入出力通信方式の詳細化、本方式の実装と実機による評価、シミュレータによる他方式との比較検討、クラスタ間ネットワークの検討、画像・音声インターフェイスの設計、などがあげられる。

なお、RWC-1は1996年3月に1024PE版が、入出力系を含めて完成する予定である。

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、古谷超並列・ニューロ研究部長、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WGの諸氏に感謝いたします。

参考文献

- [1] 坂井修一、岡本一晃、松岡浩司、廣野英雄、児玉祐悦、佐藤三久、横田隆史、超並列計算機RWC-1の基本構想、並列処理シンポジウムJSPP'93, pp.87-94 (1993).
- [2] 松岡浩司、岡本一晃、廣野英雄、横田隆史、堀敦史、児玉祐悦、佐藤三久、坂井修一、超並列計算機RWC-1における記憶構成、電子情報通信学会CPSY (1993)
- [3] 横田隆史、松岡浩司、岡本一晃、廣野英雄、堀敦史、児玉祐悦、佐藤三久、坂井修一、超並列計算機RWC-1の相互結合網、情処計算機アーキテクチャ研究会SWoPP93発表予定 (1993).
- [4] Thinking Machine Corporation, The Connection Machine CM-5 Technical Summary, 1991.
- [5] Intel Corporation, Paragon XP/S, Product Overview, 1991.
- [6] nCuber Corporation, nCube3: Technical Overview, 1992.
- [7] Parsytec, Technical Summary Parsytec GC, Ver.1.0, 1991.