計算機アーキテクチャ 101－4
（1993．8．19）

超並列計算機 RWC-1 の相互結合網

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†

*新情報処理開発機構 超並列三菱研究室
†新情報処理開発機構 つくば研究センタ
‡電子技術総合研究所

1,000 台クラスの規模を予定している超並列計算機プロトタイプ RWC-1 の相互結合網について検討する。RWC-1 上で行なわれる並列処理の方式に適合させるため、まず相互結合網に求められる要件を整理する。そして circular Omega をベースに拡張した新しい相互結合網 multi-plane circular-Banyan ($(CB)^n$), Cube-Connected circular-Banyans (CCCB) を提案する。これらの網は次数、直径ともに小さく、そのうえ良好な転送特性を持つため、要件をほぼ満足する。また本稿では超並列マシンの時分割運用についても触れ、網上に浮遊するパケットを回収するための効率良い方法を示す。

Interconnection Networks for the Massively Parallel Computer RWC-1

Takashi Yokota* Hiroshi Matsuoka† Kazuaki Okamoto† Hideo Hirono†
Atsushi Hori† Yuetsu Kodama‡ Mitsuhisa Sato‡ Shuichi Sakai†

*Massively Parallel Systems Mitsubishi Lab., Real World Computing Partnership
†Tsukuba Research Center, Real World Computing Partnership
‡Electrotechnical Laboratory

In this paper, we discuss interconnection networks (INs) for a massively parallel computer RWC-1. Since this machine employs a novel processor architecture 'RICA', which leads to efficient fine-grain parallel processing, we first discuss requrements for INs, and then, propose some novel INs named multi-plane circular-Banyan ($(CB)^n$) and Cube-Connected circular-Banyans (CCCB). These INs have low degrees and a short diameter. Evaluation results show their desirable characteristics. We also point out a problem at time-sharing use of multiprocessors, and propose an efficient packet-collection mechanism.

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)研究計画の一環として、超並列計算機の検討・開発を進めている。その開発を通して汎用超並列計算機のためのアーキテクチャ／ソフトウェアを確立するとともに、RWCにおける新しい情報処理の実行母体となるマシンを構築することを目標としている。RWC-1は、その最初のプロトタイプに位置付けられる超並列計算機であり、プロセッサ数1,000台の規模を予定している[4]。RWCのさまざまな並列処理方式に対応するため、特に細粒度並列処理を重視しつつ、汎用性に留意しながら、RWC-1アーキテクチャの検討を行なっている。

一般に並列計算機では、プロセッサ間の通信が本質的であり、特に多数のプロセッサを搭載する超並列計算機では避けて通れない問題である。RWC-1では、坂井らの提案による計算と通信を融合した新しいプロセッサアーキテクチャ **RICA** (Reduced Interprocessor-Communication Architecture, [3]) を採用することにより、通信の問題の解決を計っている。これは、これまで計算とは分断して捉えられることの多かった通信を計算と一体化することにより通信コストを下げ、また、効率の良い細粒度並列処理を可能にするものである。RICAの採用により、RWC-1では、他の(超)並列計算機では困難であった処理形態が可能になり、超並列処理方式に新たな道を開くものと期待される。

このようにRWC-1ではRICAを採用することが中心的な特徴のひとつとなる。相互結合網はRICAにより直接定義されているわけではないが、処理のスタイルが従来の(超)並列計算機とは大きく変わってくるため、これに合わせて新たに検討し直す必要がある。またその一方で、RWC-1では超並列環境でのOS／言語のあり方を追求する目的もあり、相互結合網はこれらからの要求にも応える必要がある。

本稿では、まず、RWC-1に用いる相互結合網の要件を整理する。そして、その要件を満足する新しい相互結合網を提案し、ルーティング方式から基本的な特性の評価結果までを紹介する。さらにマシンの時分割運用のためのサポート機能を取り上げ、RWC-1で実装を予定している方式を提案する。

2 RWC-1 相互結合網の要件

2.1 RICA に基づくノード構成

RICA (Reduced Interprocessor-Communication Architecture) は、RISCを並列処理向きに拡張し、単純で高速な通信の機能をプロセッサアーキテクチャへと統合したものであり、

- メッセージのハンドリングコストを下げ、並列処理の本質的な問題といえる通信コストの問題を大幅に緩和する
- RISCにより逐次実行の効率を維持する

の2点をねらいとしている。

RICAの概念図を図1に示す。到着したパケットは、結合網から直接RISC実行部に注入され、すぐにそこで処理される。これはハードウェアによって自動的に行なわれ、一切のソフトウェア・オーバーヘッドを生じない。また、専用のパケット送出命令を備えており、RISC実行部のデータパスから直接に送出するため、パケット送出のハンドリングコストもほとんどゼロに抑えられる。

このようにRICAでは通信と処理が一体化されるため、PE (Processing Element) 間通信には、データをバルク化して送るような粒度の粗いものばかりでなく、細かなメッセージを多数送るような粒度の細かいものに有利である。

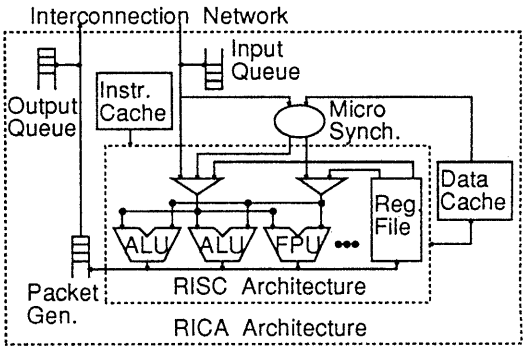

図 1: RICA: Reduced Interprocessor-Communication Architecture

RWC-1の相互結合網を検討するにあったては、まず、RICAに基づく並列計算機での通信の特性について考慮しなければならない。そこで以下の項目が相互結合網の要件として求められる。

1. 十分なバンド幅、高スループット
   パケットはRISC実行部から直接送出され、到着後、送り先のRISC実行部において直接に消費される。このため、結合網のバンド幅がPEでのパケット生成・消費のレートよりも小さい場合、RISCパイプラインのストールを招き、実行効率を落すことになる。

RWC-1 プロセッサは、データパス 64 ビット幅、100 MHz 動作を設計目標としており、パケットの生成・消費速度は最大 800 MB/s になる。PE のパイプラインをストールさせないためには、結合網はこの転送バンド幅を確保する必要がある。
さらに、この転送レートを十分に活かし、高効率な処理を持続できるスループットも必要である。

2. 十分に短いレイテンシ
細粒度の並列処理を行なう場合は、メッセージ転送のレイテンシが問題になりやすい。パケットのハンドリングコストは RICA の採用によって大きく減るため、結合網内の通過に要する時間の比重が高まる。このため網のレイテンシは十分に低く抑えなければならない。

3. 超並列マシンを実現できるスケーラビリティ
上記の項目を満足しつつ、使用可能な技術のレベルで実際に製作できることが必要である。たとえば、理想的な特性を備えていてもノードの次数の大きなトポロジーは容認されない。
また、一般に相互結合網は、規模が大きいほど通信 1 回あたりのレイテンシやスループットなどの特性が相対的に悪化するが、超並列を目指すシステムでは、この悪化を最小限に抑えなくてはならない。

2.2 RWC-1 の運用面からの要求

高価な超並列システムは、できる限り無駄なく運用することが求められる。この要求は、マシンの並列規模が大きくなるほど厳しくなるものと予想されるため、超並列マシンでは必須と考える。

RWC-1 はスタンドアロンで動作し、マルチユーザ・マルチプロセス環境を提供する。ユーザは RWC-1 マシンの一部 (あるいは全体) を使って応用プログラムを実行する。同時に複数のユーザが使用することも許される。

また、RWC-1 で想定している応用プログラムの中には、リアルタイム性を要求するものがある。たとえば、音声認識などがその典型例である。マシンの運用に際しては、このようなリアルタイムプロセスの存在も考慮しなければならない。

2.2.1 空間分割

並列マシンの一部に並列実行プロセスを割り当てることにより全体を複数ユーザで共有する空間分割は、すでに多くのマシンで実現されている。我々は分割領域 (パーティション) の特性として

- 閉 (closed) パーティション
- 開 (open) パーティション

の 2 つを定義している。同一パーティションに属する PE 同士の通信が、他のパーティションでの通信と干渉せず、完全に独立しているものが閉パーティションであり、干渉しあうものが開パーティションである。

開パーティションでは通信の干渉があるため、明らかにリアルタイム性を保証することができない。したがって RWC-1 の結合網では閉パーティションを実現しなくてはならない。

2.2.2 時間分割

RWC-1 では複数プロセスの時分割実行も要求されている。この機能は一部の商用計算機でもすでに実現されているが、マシンをインタラクティブに使用する場合のレスポンス時間や、リアルタイムプロセスの事を考慮し、できるだけオーバーヘッドの少ない高速なものをアーキテクチャ上で実現する必要がある。

RWC-1 では、時間分割はパーティションを単位とし、パーティションごとに独立に行なわれる。プロセスのリアルタイム性が損なわれるのを防ぐため、時分割のタイムスロットにまたがる通信の影響は避ける必要がある。また、時間分割と同時に、パーティションの統合や再分割が動的に行なえることが重要である。

3 RWC-1 相互結合網の提案

3.1 circular-Banyan のクラスの結合網

上に挙げた要件をもとに、RWC-1 の相互結合網として具体的にどのようなものが適しているか考えてみる。

スケーラビリティの観点から結合網の次数はサイズによらず一定でなくてはならず、しかも、バンド幅を広くとる必要があるために、次数の大きなものは採用できない。また、通信の粒度が小さいため、結合網の各リンクを双方向にすると、方向切替えの時間コストがきわめて大きくなると予想されるため、現実的にはリンクはすべて単方向となる。すなわち、結合網の次数は単方向リンクのポートの数で数えなければならない。

結合網の特性のみを見れば、間接多段網 (オメガ網など) をはじめとして、CM-5 等で採用されている Fat-Tree、HyperCrossbar など、有望と思われるものが多いが、これらは超並列計算機に適用する場合、結合網のハードウェアコストが高くなり、RWC-1 にとって必ずしも最適とはいえない。

既存の直接網の中で、前項に挙げた要件を最も良

く満足するものとして、高並列計算機 EM-4 で採用された circular-Omega[5] が挙げられる。circular-Omega は、間接多段網であるオメガ網の各スイッチの位置にプロセッサを割り当てる形で直接網化したものである。パケットは間接オメガ網と同様に左→右の一方向に流れる。右端の段の出力は左端の入力にラップアラウンドされる。各ノードが入力 2 ポート＋出力 2 ポートで構成されるため次数が小さく、その上、網の直径が $O(\log N)$ (PE 台数は $N \log N$) で抑えられる特徴を持つ。

ただしオリジナルの circular-Omega では閉パーティションを構成しにくいため、ここではバンヤン網をベースとした circular-Banyan に注目する。

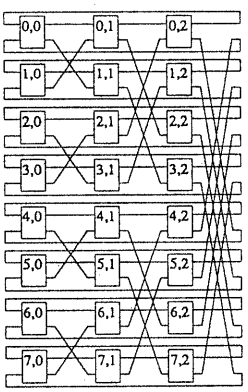

図 2: circular-Banyan の構成

ルーティング. circular-Banyan では、図 2に示すように、数個 ($\log_2 N$ 個) の PE が横方向に平行なリンクで接続され、ひとつの単方向リングを構成する。これを circular-Omega にならい、グループと呼ぶ。各 PE はグループアドレス (GA) と桁アドレス (CA) の組でアドレス付けられる。

circular-Banyan のグループ間接続 (クロスリンク) は、PE (GA, CA) から PE $(GA', (CA+1) mod S)$ に接続される。S は結合網の桁数であり、$GA' = brev(GA, CA)$ である。$(brev(n,i)$ は n の i 番目のビットを反転することを示す)。

このことから、PE (GA, CA) から (GA', CA') へのセルフルーティングは、たとえば次のように行なわれる[1]。

1. $GA' = GA$ かつ $CA' = CA$ ならば、パケットは当該プロセッサに送る。
2. GA, GA' の第 CA-th ビットが等しくなければ、クロスリンクに転送する。
3. それ以外の場合、平行リンクに転送する。

すなわち、パケットはグループアドレスが一致するまでグループ間を渡り歩き、一致した後、目的の桁アドレスまで平行リンクを進む。

空間分割. circular-Banyan では、グループを単位として空間分割が可能である。閉パーティションとするために、リングを構成しているグループは分断しない。また、パーティションに含まれるグループの数が 2^n (n=0, 1, 2, ...) に限られるなどの制限がある。

circular-Banyan での空間分割例を図 3に示す。基本的には、パーティションを切る際、PE のグループアドレス (GA) の上位ビットを等しくするだけで良い。この上位ビットに対応するクロスリンク (図中破線) が使用されなくなり、閉パーティションが実現される。

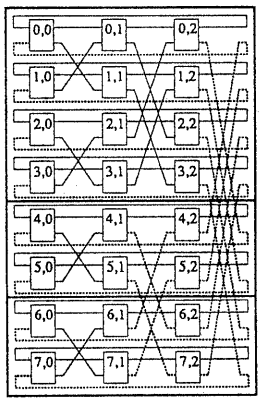

図 3: circular-Banyan での空間分割例

3.2 circular-Banyan の拡張

3.2.1 $(CB)^2$

上述のように circular-Banyan の特徴は、小さい次数でありながら網直径を小さくできる点にある。しかしながら、超並列計算機 RWC-1 に適用しようとする場合を考えると、改善の余地がある。

最も大きな問題は、ひとつのグループで構成されるリングが大きくなるために、小規模なパーティションでの通信特性が悪化することである。たとえば 1,024 PE すなわち 8 PEs/group × 128 groups の構成で 16 PE のパーティションを作る場合を考える。そのパーティションには上下に隣接した 2 つのグルー

[1] 複数通りのパスを持つ場合が存在するため、唯一のアルゴリズムではない。

プが割り当てられるが、その2つのグループの間は1ペアの単方向リンクでしか結ばれていないため、そのグループ間リンクが隘路になってしまう。

そこで、上記の欠点を補い、さらに、大規模化したときの相対的な特性の低下を補うこともねらって、ルータの入出力ポートを増し、circular-Banyan を拡張することを考えた。そのひとつの解が $(CB)^2$ であり、以下にその概要を説明する。

トポロジー. $(CB)^2$ は、小規模な circular-Banyan でクラスタを構成し、クラスタ間の接続に Banyan 接続を適用するものである。その構成例を図4に示す。図中、破線で示したリンクは2つの平面の Banyan 接続で共有されていることを示す。

各ノードは3入力×3出力のポートを持つ。そのうち入出力各2ポートをクラスタ内接続用に用い、上記の circular-Banyan を構成する。ここで、入出力各1ポートが未使用のまま残るが、これをクラスタ間の接続に用いる。その接続のしかたは circular-Banyan のクロスリンクと同じ方法で行なわれる。すなわち、PE のクラスタアドレスを XA として、PE アドレスを XA, GA, CA の組で表現するとき、PE (XA, GA, CA) から出るクラスタ間リンクは、PE $(brev(XA, CA), GA, (CA+1) mod S)$ に接続される。$brev(n, i)$ は上と同様に、n の第 i ビット目の反転を表す。

この結合網は、S 個の PE で構成されるグループ(リング)を2次元的に $2^S \times 2^S$ 個並べ、各次元ごとに Banyan 接続したものと説明することができる。すなわち (GA, CA) 平面で見ても Banyan 接続になり、また、(XA, CA) 平面においても同様の Banyan 接続になる(ただしグループ内でリングを構成する平行リンクは両平面で共有される)。同様の考え方でさらに次元数を拡張することが可能であり、一般にこの方法で作られた結合網を multi-plane circular-Banyan ($(CB)^n$) と総称する。$(CB)^2$ はそのひとつの実現例である。

ルーティング. $(CB)^2$ のルーティングは、circular-Banyan のものを素直に拡張すればよく、次のように表される。

1. $XA' = XA, GA' = GA, CA' = CA$ の場合は当該 PE に送る
2. XA, XA' の第 CA-th ビットが等しくなければクラスタ間クロスリンクに転送
3. GA, GA' の第 CA-th ビットが等しくなければグループ間クロスリンクに転送
4. それ以外の場合、平行リンクに転送

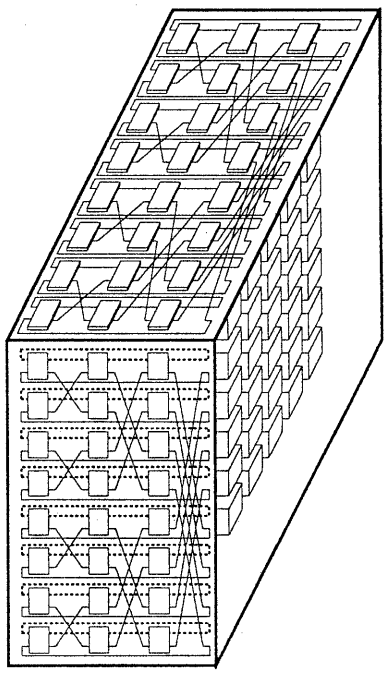

図 4: 8×8×3 $(CB)^2$ の構成

3.2.2 CCCB— Cube-Connected Circular-Banyans

ここで $(CB)^2$ のルーティングを見直してみる。各ノードからは、GA を変えるグループ間クロスリンクと、XA を変えるクラスタ間クロスリンクの両方が出ている。パケットをどちらのリンクに転送しても良い場合、上記のルーティングでは、まずクラスタ間クロスリンクに出し、そのパケットが一周して同じ桁位置に戻って来たところでグループ間リンクに出すことになる。このため、パケットが目的のグループに到達するまでの周回数が多くなり、平行リンクを使う頻度が相対的に高くなる。結果として網全体の性能が負荷の高い平行リンクに制限されてしまう。

以上のような不都合を回避するため、$(CB)^2$ を変更し、クラスタ間リンクでは桁位置を変えないようにする。図5にその例を示す。

これは、同一の (XA, CA) 平面上にあるグループ(= リング)をハイパーキューブ形に接続したものであり、この平面上において、Cube-Connected Cycles (CCC, [2]) のサブセット(リング内が単方向)になっている。したがって、全体としてみれば、Cycle 間をキューブ接続した CCC と同様の拡張方法を

用いて、circular–Banyan をキューブ形に接続したものとみなすことができる。この結合網は Cube-Connected Circular–Banyans(CCCB) と呼ぶことにする。

CCCB ではクラスタ間リンクで桁位置を変えないため、同一の桁内でクラスタ、グループとも変えることができる。すなわち、パケットは桁位置が一周する間にクラスタ、グループとも目的の位置に到達できる。あとは目的の PE まで平行リンクを進めばよい。

CCCB のルーティングは、$(CB)^2$ のものをそのまま使うことができる。

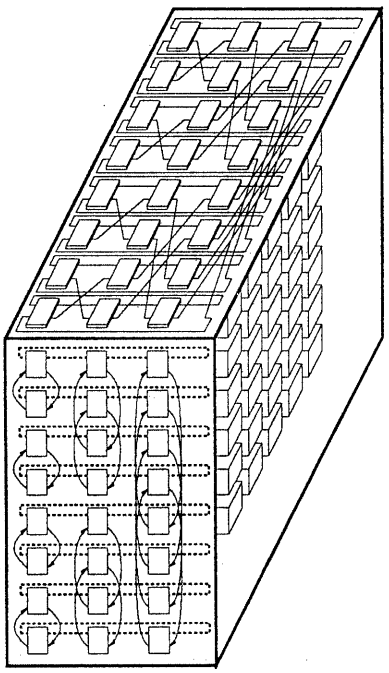

図 5: 8×8×3 CCCB の構成

3.3 ストアアンドフォワード・デッドロックの回避

circular–Omega では、通信路が単方向で環状になっていることを利用して、らせんバッファ手法によりストアアンドフォワード・デッドロックを防いでいる。パケットが特定の桁を通過するときに、ルータ内でパケットが保存されるバッファのクラスを 1 だけ上げる。circular–Omega(Banyan) の場合、最悪 2 周以内に目的の PE に到着するため、必要なバッファクラス数は 3 になる。

上に示した $(CB)^2$ や CCCB は、いずれも circular–Omega で考案されたらせんバッファ手法を適用することができる。ただしパケットが目的 PE に到着するまでに、$(CB)^2$ で最悪 3 周かかり、CCCB では 2 周かかる。このため、必要なバッファクラス数は $(CB)^2$ で 4, CCCB で 3 となる。

3.4 基本特性の評価

RWC–1 相互結合網の検討のために結合網シミュレータを作成し、さまざまなトポロジーやルーティングについて評価している。

以下のようなシミュレーション条件

- PE 総数は 1,024 台。
- パケット長 = 3 ± 1 [ワード]
- virtual cut-through ルーティング。バッファクラスはトポロジーにより最低限必要とされる数のみ。
- 入力ポートごとにバッファクラス数のバッファを設ける。各バッファの容量は 16 ワード。

を統一したうえで、通信パターンとして

- 一様ランダム通信
- 1/4 パーティション通信
  全体を 4 つに等分割し、パーティション内で一様ランダム通信を行なう。
- ホットスポット通信
  5% のパケットが特定 PE に行く。他の 95% のパケットは一様ランダムに行き先が決められる。
- メッシュエミュレーション通信
  32 × 32 メッシュのエミュレーション。4 近傍に送出し、その 4 近傍からのデータが揃うまで待つことを繰り返す。

の 4 通りを設定して、スループットおよびレイテンシを評価した。

その結果を図 6, 7, 8, 9に示す。各図中、CB は circular–Banyan を表し、Adaptive CB はルーティングに適応制御を行なった circular–Banyan を表す。

図 6, 7は、シミュレーション結果より算出した実効パケット送出間隔と平均レイテンシとの関係を示している。左側のグラフが立ち上がった部分で最短送出間隔が読み取れ、右側のなだらかな部分から網が空いているときのレイテンシを読むことができる。$(CB)^2$, CCCB は、間接オメガ網には及ばないものの、良好な通信特性が得られることがわかる。

図 8, 9は、スループットと平均レイテンシのプロットである。ホットスポット時 (図 8) のスループットはメッシュ／トーラスと同程度である。両者の間では平均レイテンシに差があるが、これはメッシュ／トーラスでは tree saturation の影響の少ない部分

が残ることによるものと考えられる。メッシュエミュレーション時(図9)は、物理的にメッシュ型の結合を持つものには及ばないが、スループット、レイテンシとも(一般に予想されるよりも)特性の悪化を少なく抑えられることがわかる。

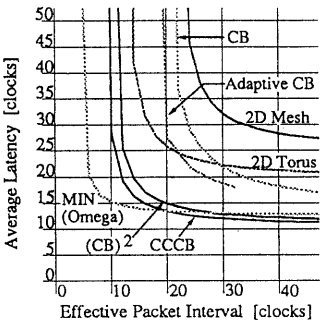

図 6: 一様ランダム通信パターンでの通信特性

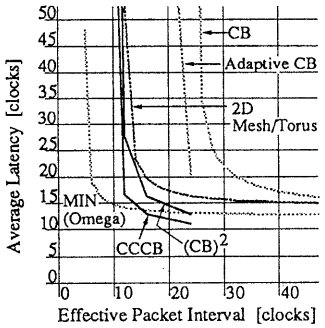

図 7: 1/4 パーティション内ランダム通信での特性

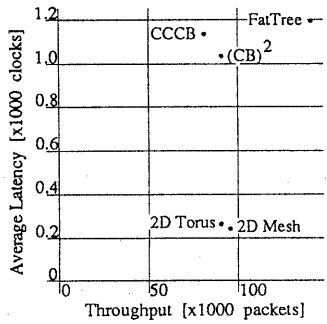

図 8: ホットスポット通信特性

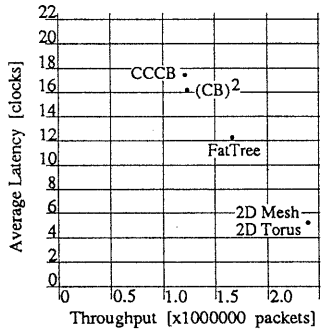

図 9: メッシュエミュレーション時の通信特性

3.5 $(CB)^2$, CCCB の構成法 — バリエーション

上では $(CB)^2$, CCCB の構成として、グループ数 = クラスタ数 = N で、桁数 = $\log_2 N$ 、PE 総数 = $N^2 \log_2 N$ の場合を示した。これは $(CB)^2$, CCCB の基本的な構成方法であるが、要求される PE 台数に応じてある程度自由に変えられる。たとえば、同一の circular–Banyan を横方向に 2 つつなげた形にして桁数を 2 倍にし、基本形の倍の PE 数を実装することができる。また、クラスタ数を半分にすれば基本形の 1/2 の PE 数となる。これはクラスタによりパーティションを切った場合と等価であり、該当するクラスタ間リンクが使われなくなるが、このリンクを平行リンクにして桁間を結ぶなど、有効な利用法が選択できる。

4 時分割運用のサポート機能

4.1 結合網上に浮遊するパケットのドレイン／フラッシュ

パーティション内の全 PE が同時にプロセスを切替えても、結合網上に浮遊しているパケットが残る。そうした浮遊パケットはいずれ目的の PE に到着するが、そのパケットを消費すべきプロセスがすでにプロセス切替えにより非活性になっているため実行できない。

第 2 節で述べたように、RWC-1 ではリアルタイムプロセスへの考慮が必要なことから、タイムスロットにまたがっての通信の影響は避けなければならない。したがって網上に残ったパケットは、プロセス切替の際にカーネル管理下でいったん網外に保存してサスペンド状態にしておかなくてはならない。したがってプロセス切替えのオーバーヘッドを抑えるには結合網上に浮遊しているパケットを効率良く保存するための機能が求められる。結合網自体には、

こうしたパケットを保存するための資源がないため、パケットはいったんPEを経由して主記憶上に保存されることになる。

ここで、浮遊パケットをPEに送る方式として

- ドレイン (Drain)
- フラッシュ (Flush)

の2つを定義する。フラッシュはパケット内に書かれている目的PEに届けるものであり、ドレインは最も近いPEに送るものである。フラッシュならば、保存してあったパケットをプロセスの再起動時に自PEでそのまま消費すれば良いが、ドレインの場合は保存パケットを再び結合網上に戻さなくてはならない。

しかし、フラッシュでは網上の全パケットが目的地に着くまで待たねばならず、さらに、網上にパケットがなくなったことを検出するために大きなコストがかかる。これに対してドレインは最悪ルータ内の全バッファ容量を保存するだけの時間ですむ。このためRWC-1では次に述べる方法によりドレインをサポートする。

4.2 ドレインの方法

1. ドレインの起動
   パーティションに属する全PEに対して、ドレイン起動の指示がブロードキャストされる。この指示は、これまでに述べたデータ転送用の結合網(主結合網)とは別個に実装される入出力用の結合網[1]から流される。これは、主結合網が輻輳している場合でも迅速に配送する必要があるためである。また、入出力網のほうが放送に適した構造になっていることも大きな理由である。
2. ルータ間での同期
   フラッシュ開始の指示が与えられると、ルータは、出力を凍結する。ただし出力途中のパケットは最後まで出力する。出力が凍結されたら、各ポートの制御信号を使って、次段のルータ(の入力ポート)にその旨を伝える。ルータは出力ポートがすべて凍結し、また、入力ポートもすべて凍結されるまで待つ。凍結を伝えられた入力ポートは、"busy"状態を示す。ここで、全ルータでの同期は必要はなく、隣接ルータ間のローカルな同期だけで良い。全ポートが凍結されると、ルータにはそれ以上のパケットが転送されないことが保証される。
3. ルータ内パケットの掃き出し
   この時点でルータは直接接続されているPEに通告し、ルータ内に残っていた浮遊パケットがPE経由で保存される。
4. ルータ凍結の解除
   浮遊パケットの掃き出しが完了したら、切替え後に起動されるプロセスの前回保存パケットを回復する。そしてルータは入力ポートをパケット受信可能状態("ready")にし、出力ポートの凍結を解除する。これにより、ドレイン終了の同期操作も自動的に行なわれる。

5 おわりに

現在、研究・開発が進められている超並列計算機RWC-1の相互結合網について、求められる要件を整理し、それを満足するものとして、circular-Banyanをベースとした新しい結合網$(CB)^2$, CCCBを提案した。シミュレーションの結果から、これらの結合網が良好な通信特性を持ち、RWC-1の相互結合網として妥当であることが示された。また、超並列マシンの時分割運用を効率的にサポートするためのドレイン機能を提案した。

今後、特性評価を進めるとともに、機能などの仕様も固め、RWC-1の相互結合網として実装する予定である。

謝辞 本研究を進めるにあたり、有形無形の支援をいただいているRWCつくば研究センタの島田潤一所長ならびに古谷立美 超並列・ニューロ研究部長、$(CB)^2$のヒントをいただいた慶応大学の天野英晴講師、討論に加わっていただき有益な示唆をいただいたRWC超並列アーキテクチャWSのメンバ諸氏、電総研およびRWCつくば研究センタの各位に厚く感謝します。

参考文献

[1] 廣野 英雄ほか. 超並列計算機RWC-1における入出力機構. 情処学アーキテクチャ研究会, ARC-101-5, 1993.

[2] F. P. Preparata and J. Vuillemin. The Cube-Connected Cycles: A Versatile Network for Parallel Computation. *Communications of the ACM*, Vol. 24, pp. 300–309, May 1981.

[3] 坂井 修一ほか. 超並列計算機の原理. 信学技報, CPSY92, 1992.

[4] 坂井 修一ほか. 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.

[5] Shuichi Sakai, Yuetsu Kodama, and Yoshinori Yamaguchi. Design and implementation of a circular omega network in the EM-4. *Parallel Computing*, Vol. 19, No. 2, pp. 125–142, 1993.