並列オブジェクト指向トータルアーキテクチャ A-NET のためのトポロジ独立なルータの構成

吉永 努[†] 茂木 久[††]
佐々木 昌[†] 馬場敬信[†]

並列オブジェクト指向実行モデルに基づく A-NET 高並列計算機のルータを設計した．A-NET ルータは，静的に可変な種々のネットワークトポロジをサポートするため，プログラマブル通信制御装置を用いてメッセージの経路選択を行う．メッセージは，適応型バーチャルカットスルー方式による可変長パケット交換により実現する．また，ユーザ定義のオブジェクトやメソッドを動的にノード間転送するために，サーキットスイッチ方式のデータ転送もサポートする．ルータは，ホスト用に1ポート，および隣接ルータとの接続用に6ポートを持ち，PE インタフェース回路，メッセージセンダ／レシーバ，パケット退避用バッファなどをクロスバ網で接続した構成をとる．これらクロスバ網に接続した各ブロックは，パケットやオブジェクトコードなどの転送主体となり，パケットの経路選択，およびクロスバ網の設定はプログラマブル通信制御装置が行う．各ブロックは，それぞれステートマシンを内蔵して独立に動作し，ルータ内で並列にデータ転送が行える．通信性能を評価した結果，無衝突時のパケットの1ホップ当りの経路選択時間は，約 2 μs で，平均的なサイズのパケットの転送時間は 10〜48 μs 程度（距離 1〜20）であることが分かった．この値は，PE 上での1メッセージ当りのユーザプログラムの連続実行時間とバランスの取れたものといえる．

Organization of a Network-Topology Independent Router for a Parallel Object-Oriented Total Architecture A-NET

Tsutomu Yoshinaga,[†] Hisashi Mogi,[††] Syo Sasaki[†]
and Takanobu Baba[†]

We have designed a router for a parallel object-oriented machine. The A-NET router provides two functions, that is, message routing and dynamic object allocation, in a network-topology independent fashion using a programmable communication controller. It supports both adaptive virtual cut-through packet switching and the circuit-switching transfer of an object code. It consists of several hardware blocks, a message-sender, a message-receiver, a PE interface circuit, a packet buffer, 6 ports to connect with other routers, and a port for a host computer. These blocks have their own state machines and are connected to a crossbar network, enabling them to exchange data simultaneously. According to the preliminary evaluation, the time to decide a route per hop is about 2 μs, assuming that there is no conflict. The propagation delay of an average size message traveling 1 hop is 10 μs, and it takes 48 μs for one traveling 20 hops. These times are comparable with a typical continuous execution time of user programs per message.

1. はじめに

超並列マシンの1方式として，メッセージ交換型の MIMD マルチコンピュータが注目されているが[2]，その相互結合網に大きく2つの実現法がある．1つは，固定された結合網上でメッセージの経路選択に決定性のアルゴリズムを用いることにより，1ホップ当りのメッセージの通信遅延を極力小さくするものである．この方式を用いた J-Machine[5]，Prodigy[15]，AP-1000[9] などでは，ワームホールーティングにより距離に大きく影響されない通信遅延を達成している．2つ目は，静的，あるいは動的にネットワークトポロジを可変にする[16]と共に，各計算ノードへのタスク割り付けを最適にして，より柔軟にハードウェアとソフトウェアを適合させようとするアプローチである．この方式を用いたものに Computing surface や Parsytec GC などがある[13]．

† 宇都宮大学工学部情報工学科
Department of Information Science, Faculty of Engineering, Utsunomiya University
†† 東芝総合研究所
Research & Development Center, Toshiba Corporation

前者は，ネットワークトポロジとメッセージのルーティングアルゴリズムを限定することにより，通信チャネル当りのバッファ容量を抑えながら高い通信能力を得ることができるが[6]，負荷の重いランダム通信においてはスループットが上がらないなどの問題点も指摘されている[12]．このため，負荷の集中を避けるために，局所的な通信状況に従って経路を選択する適応型のルーティング手法が数多く提案されている[8),17)]．CM-2では，動的にネットワークの負荷調整を行うルータ機構を用意しているが，オーバヘッドが大きいため，隣接ノード間についてはNEWS通信や直接ハイパーキューブアクセスを用意している[7]．

しかしながら，ネットワークトポロジを1つに限定した場合，種々のアプリケーションプログラムの持つ構造を直接的に反映することは困難になる[14]．その結果，通信距離が伸び，ブロッキングも起こりやすくなって，ネットワークの性能を生かすことが難しくなる．これらのことから，必要に応じて結合形態を選択できるマルチコンピュータの構成要素プロセッサの開発が期待される．

このような背景から，我々は，ネットワークトポロジ独立な高並列計算機用の専用ノードプロセッサの開発を進めている[1),3),20)]，本研究の特徴は，並列オブジェクト指向を核概念として並列プログラミング言語，およびその実行環境を統合的に検討している点にあり，ハードウェアアーキテクチャも並列オブジェクト指向プログラムの効率的実行を目的に設計している．対象とする応用問題は，比較的粒度の大きな非構造・非定型な問題領域である．並列オブジェクト指向言語による問題の記述とその効率は文献18)などにまとめられている．

本稿では，A-NET計算機においてメッセージパッシングを支援するルータの設計方針とハードウェア構成について述べた後，ルータの性能予測を行い，ノード内処理と通信時間のバランスなどについて考察する．

2. ルータの設計方針

ルータの設計方針は，次の3点に要約される．

（1） ネットワークトポロジ独立

A-NETでは，幅広い応用分野に柔軟に対応できることを重視して，ネットワークが再構成可能な専用ノードプロセッサにより高並列計算機を実現する．また，ベースとする並列オブジェクト指向実行モデルは，メソッドを内蔵する独立性の高いオブジェクトがメッセージパッシングにより自立的，かつ並列に動作することを基本としているため，ノード間には共有メモリを置いていない．

ネットワークをトポロジ静的可変にするか動的可変にするかは性能／コストによって決まるが，プロトタイプにおいては，ハードウェア量を考慮して静的可変というアプローチを採ることにした．これにより，言語処理系は指定された物理プロセッサグラフにプログラムの内包するタスクグラフを最適にマッピングすることができる[4]．A-NETの記述言語であるA-NETLでは，同一のメソッドを持つ多数のオブジェクトを，言語処理系に静的に生成させるインデックスト・オブジェクト[19]をサポートしている．したがって，動的オブジェクト生成は特にプログラムに柔軟性が必要な時に限定することができ，プログラム記述の簡便さ，実行時間の短縮を図ることができる．

以上のことから，ルータにはネットワークトポロジに応じたメッセージの経路選択と動的オブジェクトの割り付け機能が必要となる．

（2） オブジェクトコード転送とメッセージ通信の両立

オブジェクトレベルの並列性を引き出すためには，メソッドに相当する実行コードも各ノードに独立して持つと共に，遅延の少ないメッセージ通信を達成する必要がある．A-NETルータでは，動的オブジェクト生成に伴うオブジェクトコード転送と通常のメソッド起動のためのメッセージ転送を区別して2種類の通信方式をサポートする．これらを単一の方式で実現することも可能であるが，それぞれの発生頻度，およびデータ転送量が大きく異なるため，別々の方式を用いた方が効率的であると判断した．通信機能については，次章で詳しく述べる．

（3） ノード内／ルータ内並列動作の実現

メッセージのオーバヘッドを減らすため，メッセージの送信元，および中断ノードではPEの動作を中断することなくルータがメッセージを送受信する．ただし，宛先ノードにおける受信時には，ローカルOS[21]のオーバヘッドを考慮して，ルータがPEの動作を一時停止させてPE内のローカルメモリに直接メッセージを書き込む方式を採用する．

また，ルータ内も独立して動作する複数のブロックからなる構造とし，できるだけ各ブロック間でのデータ転送を並列に行えるものとする．

3. 通信機能

2章でも述べたとおり，A-NET ルータは基本的にオブジェクトコード転送とメッセージ転送の2種類の通信機能を実現する．以下に，それぞれについて具体的に述べる．

3.1 オブジェクト転送

3.1.1 オブジェクト転送が必要な理由

逐次的なオブジェクト指向言語においては，クラスにインスタンスの生成メッセージが送られて動的にオブジェクトが生成されるが，一般にメソッドコードは共有される．しかしながら，オブジェクトが本来備えている自立的な計算能力を活用するためには，分散メモリ型マルチコンピュータの各ノードにメソッドコードを持つオブジェクトを分散させて並列実行すべきである．メソッドを各ノードに分散させるためには，そのメソッドを実行する可能性のあるノードには，あらかじめオブジェクトコードを割り付けておくか，動的オブジェクト生成時にコード転送をする必要がある．このことは，共有メモリを持つクラスタ構造のネットワークを採用したとしても，クラス外に動的オブジェクトを割り付ける可能性がある限り基本的に変わらない．

A-NET では，二項演算やリスト操作などのデータ操作はマイクロコードにより実行し，また算術関数や文字列操作などは必要に応じてライブラリオブジェクトとしてあらかじめノード内に割り付けられた命令列を実行する．したがって，これらのプリミティブメソッドについてはコードのリモートアクセスや動的転送は必要ないが，それ以外のユーザ定義のメソッドについては，動的オブジェクト生成時に命令コードも同時に割り付け先ノードに転送する必要がある．ただし，A-NETL には同一のメソッドを持つ複数のオブジェクトを静的に生成する機能も定義されているため，ユーザは実行性能を重視した静的生成と柔軟性を重視した動的生成をプログラムで制御可能である．

3.1.2 動的オブジェクト生成の手順

動的オブジェクトの割り付けに伴うコード転送は，ルータがノード間で DMA 転送する．クラスの存在するノード（クラスノード）は，転送距離を減らすとともに，オブジェクトをできるだけ多くのノードに分散させるように割り付け先の候補を決定する．

オブジェクトコード転送にはサーキットスイッチ方式を採用しているが，これは，

1. 平均的なオブジェクトコードサイズが1KB以上と大きいこと[21]，
2. 宛先探索の返答メッセージをそのまま回線設定に利用できること，

などの理由による．これにより，ロードするデータが転送途中でブロックされ，長い時間多くの経路を塞ぐという事態を避けることができる．動的オブジェクト生成の手順は以下のとおりである．

1. クラスがオブジェクト生成用のメッセージを受信すると，割り付け先探索のためのポーリングメッセージ（POL）が送出される．一定数以下のオブジェクトが割り付けられているノードを探すため，POL は割り付け済みオブジェクト数の上限を引数として持つ．
2. POL を受信したノードは，自ノードのオブジェクト割り付け数や空き領域サイズを調べ，割り付け可能である場合，アクノリッジメッセージ（ACK）を返答し，そうでない場合，別の宛先を設定して POL を回送する（ハブポーリング）．ACK を返答する時，ルータはオブジェクト割り付けに備えて PE の動作を一時中断させる．
3. ACK を返送する割り付け先からクラスノードまでの中継ルータは，ACK 転送後，オブジェクト転送のため ACK の返送経路と逆向きに回線を設定する．
4. ACK を受信したクラスノードのルータは，ACK によって回線設定された経路を通してオブジェクトコードを DMA 転送する．
5. 転送されてきたコードを DMA 割り付けした宛先ノードのルータは，割り込みにより PE 上の OS を起動し，通常動作に戻る．

現在は，生成する動的オブジェクト数をある程度予想できる場合には，アロケータ[4]があらかじめクラスノードの回りに割り付けの余裕を持たせておき，トポロジに従って割り付け先を実行時に決定することとしている．ただし，このアルゴリズムは割り付け先をランダムに決定するものなどに容易に変更可能である．また，A-NET 計算機ではすべてのノードがクラスノードとなり得るため，ポールを送り出すだけのノードと受けるだけのノードという区別はない．したがって，ハブポーリングを行うとしてもシリアルポーリングの場合に比べてノードのロジックが増えることはない．POL がクラスノードに返ってくることもあり得るが，割り付け先の探索条件を動的に更新していくこ

とでライブロックの問題は回避できる.

3.2 メッセージ転送

PE 上のメソッドは，メッセージ駆動方式により実行されるが，ノード間のメッセージ交換は，255 バイト(51語)以内の可変長パケット交換により実現する．また，これを越えるメッセージについては PE 上で分割／再構成する．これまで記述したサンプルプログラムの実行結果から，多くのメッセージが 80 バイト以下のパケットに納まることが分かっているが[3),21)]，パケット分割／再構成のオーバヘッドが大きいことなどから，プロトタイプではパケットサイズの上限を大きめの値に設定している．

パケット交換の制御には，適応型／バーチャル・カットスルー方式を用いる．この方式においては，パケットの転送経路を固定せず，複数の送出可能ポートがある場合には，その時のローカルな状況により出力ポートを決定し，ブロッキングが起きた場合にはパケットをポートからルータ内のバッファに退避する．この方式を用いた理由としては，

1. ネットワークトポロジを可変にするため，経路選択に決定性アルゴリズムを用いることはできない，
2. ブロッキング時にポートを解放できることによりデッドロック回避を容易にするとともに，適応制御によりネットワークのスループット低下を防ぐ，

などの理由による．

4. ルータの構成

図1に示すように，ルータはホスト用に1ポート，および隣接ルータとの接続用に6ポートを持ち，PE インタフェース回路，メッセージセンダ／レシーバ(以後，単にセンダ／レシーバと呼ぶ)，パケット退避用バッファなどをクロスバ網で接続した構成をとる．プログラマブル通信制御装置が，パケットの経路選択，およびクロスバ網の設定を行い，クロスバ網に接続した各ブロック間でパケットやオブジェクトコードが相互に転送される．

以下に，各ブロックの構成と機能について述べる．

4.1 プログラマブル通信制御装置

トポロジ独立な適応型ルーティングをサポートする場合，各ポートに経路選択機能を持たせることはハードウェア量の観点から困難である．そこで，ポートなどのブロックの使用状況の監視とトポロジにより変化する経路選択機能をプログラマブル通信制御装置(PCC)により実現することとした．PCC をプログラマブル・デバイスで構成することにより，柔軟性と高速性を両立させている．また，ポートやセンダからの経路選択要求を PCC で調停することにより，クロスバ網などの共有資源の調停も PCC で行うことができる．

PCC の構成を**図2**に，また PCC によるブロック制御コマンドを**表1**に示す．PCC と各ブロックは，経路情報バス，および4本の制御信号線で接続されている．経路情報バスは，表1に示したコマンド，およびパケットヘッダを通信するためのものである．また，各信号線の意味は，PCC からのブロック制御(/CNTL)，ブロックから PCC への割り込み要求 (/IREQ) とその許可 (/IACK)，およびクロスバ網からパケットを受けることができる状態か否か (AVL) である．

図3に PCC の経路選択手順を示す．

1. 各ブロックからの割り込みをラウンドロビン方式で受け付ける．
2. ポート，またはセンダからの経路選択を受け付けた場合，経路情報バスを通しパケットヘッダ2バイト (パケットの種類と宛先ノードアドレス) を受け取り，トポロジとその時の各ブロックの使用状況に応じた最短経路の選択を行う．
3. 経路選択要求ブロックとパケットの送り先ブロックとの間のパスをクロスバ網により設定する．

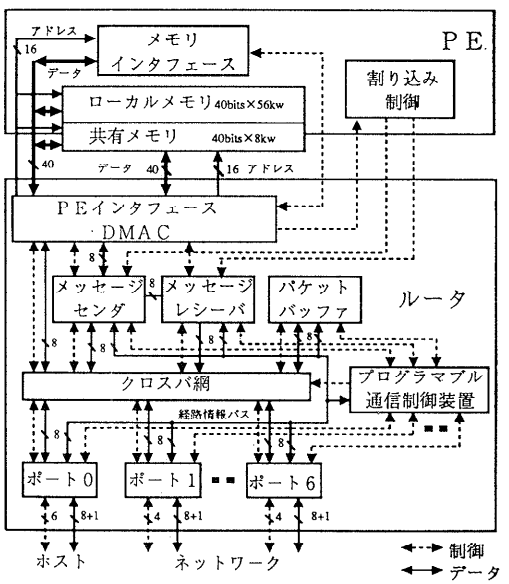


図1 ルータのハードウェア構成
Fig. 1 Hardware organization of the router.

送り先ブロックは，パケットが自ノード宛ならばレシーバ，他ノード宛ならば送信可能ないずれかのポートとなる．

4. 宛先ブロックにブロック確保コマンドを，また，要求元ブロックにパケット転送コマンドを送る．すべての出力候補ブロックが確保できない時は，それらの出力候補ブロックに対応するビットを立てた経路選択結果（ポート情報）を PCC 内に一時記憶し，次のブロックの要求を受け付ける．

ブロッキングにより転送不可となったパケットの2度目以降の経路選択は，PCC 内に記憶されたポート情報から出力候補ブロックの利用可能信号 AVL を調べ，クロスバ網を設定すればよい．2度目の経路選択で再びブロッキングを検出した場合には，ポート情報付きのパケットをパケットバッファに退避して要求元ブロックを解放する．したがって，パケットバッファからの要求は，常にポート情報からのクロスバ網設定となる．ブロッキング時に，パケットを迂回させて負荷を分散させる方法も考えられるが，ライブロックの制御が複雑化する[11]ので現在は行っていない．

また，表1で ACK 用ブロック確保／転送コマンドを通常パケット用ブロック確保／転送と別に用意しているのは，ACK 転送後にサーキットスイッチの設定が必要なためである．ACK の転送経路はそのままサーキットスイッチされるため，PCC は ACK をパケットバッファに退避することはない．このため，ACK 同士の衝突時に一方のサーキットスイッチを解除するためのポート初期化コマンドを定義している．

4.2 ポート

ルータは，4分木，3Dメッシュなど種々の構成が可能なよう，隣接ルータとの結合用に6ポートと，ホスト結合専用に1ポートを持つ．各ポートは双方向ポートであり，それらの間はデータ8ビットとパリティ1ビットの半二重リンクで結合する．半二重リンクとは，同時には両方向のデータ転送は行わず，一度には片方向のデータ転送を行うリンクという意味である．

ルータ-ルータ間のパケット交換は，受信ポート内の FIFO を緩衝バッファとして，ハンドシェークを行わず非同期的に送信側主導で行う．これは，トポロジ独立な高並列を目指すという観点から，クロックスキューの問題なども考慮し，ノード間での共通クロックは前提としていないことによる．

ポートに隣接ルータからパケットが入力されると，受信ポートはこれを FIFO に格納すると共に，PCC に経路選択要求を発する．PCC が要求を受け付けた後，受信ポートはパケットヘッダを PCC に送って，

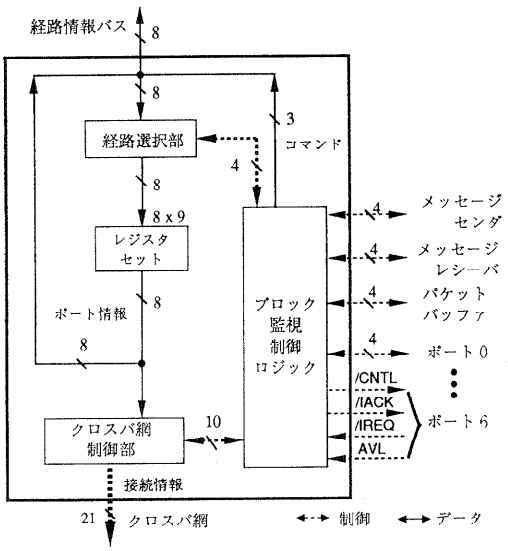


図 2 プログラマブル通信制御装置 (PCC)
Fig. 2 Programmable communication controller (PCC).

表 1 PCC のブロック制御コマンド
Table 1 Commands of the PCC for the blocks.

コマンド	制御内容
ブロック確保	クロスバ網からパケット受信
ACK 用確保	ACK 受信後サーキットスイッチ
パケット転送	クロスバ網へパケット送信
ACK 転送	ACK 送信後サーキットスイッチ
ポート初期化	ACK 衝突による初期化

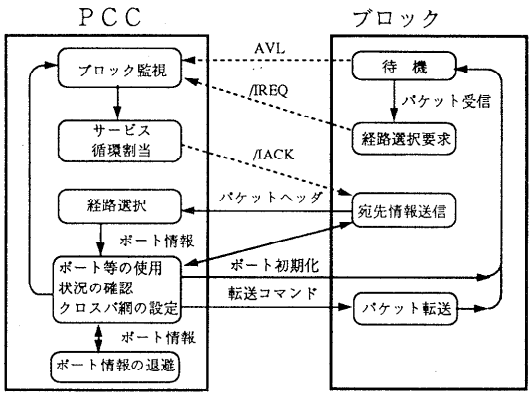


図 3 経路選択手順
Fig. 3 Routing sequence.

送り先ブロックとの間でクロスバ網が設定されるのを持つ．送り先ブロックがポートの場合，ポートが半二重であるため PCC からのブロック確保と隣接ルータからのパケット入力が競合する可能性がある．この場合には，隣接ルータからのパケット入力を優先させることで送信側主導の半二重リンクを管理する．

受信ポートは，PCC が送り先ブロックを確保してクロスバ網を設定してから，パケット転送コマンドを受け使って，FIFO 内のパケットを送出ポート，レシーバ，またはパケットバッファなどへ転送する．送出ポート内では，パケットは FIFO には格納されない．すなわち，隣接ルータの受信ポートが使用可能である時に限り送出ポートが確保される．ノード間転送中に，1つのパケットが複数のノードにまたがることも許すことにより，ストア & フォーワード方式よりも転送遅延を小さくすることができ，ブロッキングが起きなければワームホールルーティングに近い転送速度が期待できる．

ホスト-ルータ間のパケット交換は，ホスト側とのデータ転送速度の違いを吸収するためハンドシェークによりデータ転送を行う．プロトタイプにおいては，ホストの VME インタフェース回路と各ルータのホスト用ポートは競合調停用の回路を介してバス結合される．

4.3 PE インタフェース

PE インタフェース (PE I/F) は，メッセージセンダ／レシーバの制御を介して PE 用ローカルメモリ (LM)，および PE-ルータ間共有メモリ (CM) のアクセスを行う．LM については，通常は PE が専用メモリとして使用するが，OS やユーザオブジェクトのコード割り付け時と，ルータが受信した自ノード宛メッセージを PE に渡す時にはルータ側がバスマスタとなって LM を直接アクセスする．CM は，メッセージ出力キューと，LM の空き領域やメッセージポインタなど，PE とルータで共有すべき情報を保持する領域からなり，デュアルポートメモリで構成する．したがって，PE によるメッセージ送信命令の実行とルータのパケット送信処理は並列に行える．

PE I/F とセンダ／レシーバ間には，PE-ルータ間共有情報の読み書き，パケットの受け渡し，DMA 送受信などのコマンドが定義されている．例えば，オブジェクト割り付け時には ACK を送信したセンダから LM の DMA 受信コマンドを受け取って，割り付け開始番地，転送語数を設定し，PE にバスの解放を要求して DMA 受信モードに移行する．

ノード外からのパケット受信時には，レシーバからメッセージ書き込みコマンドとパケットデータを受け取って，LM へ DMA ライトする．以前は CM 上にメッセージ入力キューを設ける方式を検討していたが，メッセージが到着するまで PE はアイドル状態であることも多く，また OS によるメッセージ受信処理において，CM から LM へのメッセージコピー処理などにより実行効率が上がらないことが分かったため[21]，PE I/F により直接 LM に書き込むこととした．この機構は，メッセージ受信時に PE の動作を一時停止することを必要とするが，OS によるオーバヘッドが軽減されるため，メッセージの到着間隔と PE の内部処理のバランスが取れれば全体のスループットを上げることができる．

4.4 メッセージセンダ

メッセージセンダは，PE I/F を介して CM 上の出力キュー・ポインタを読み出し，送出すべきメッセージがあるかどうか確認する．メッセージがある場合，これを読み出して PCC にパケットヘッダを送り，経路選択を要求する．その後，PCC からのパケット転送コマンドに従って，クロスバ網にパケットを出力する手順は受信ポートと同一である．

PE 上のメッセージ送信命令は，送出すべきメッセージサイズが最大パケット長を越える時はこれを複数に分割して CM の出力キューに書き込むため，センダは通常のパケット送信処理に対して何ら特別な処理を必要としない．マルチキャストメッセージについても，PE が機械命令レベルで複製したものをセンダが連続的に送信する．この方法は，ルータでマルチキャストパケットを複製する場合に比べて多くの出力キュー領域を必要とするが，ルータでパケットの送信モードをデコードする必要がないため，ハードウェアが簡単になる．

センダは，**図4**に示すように1パケットを格納できる FIFO，読み出した共有情報やパケットヘッダを一時格納するレジスタ，メッセージポインタや共有情報の値を比較する回路，および制御ロジックからなる．また，PE I/F へのコマンドと CM のアドレス出力などはフィールド・プログラマブル・コントローラで実現する．

4.5 メッセージレシーバ

レシーバの主な機能は，ポートが受信したパケットが自ノード宛である場合，その受信処理を行うことで

ある．レシーバは，PCC からのブロック確保コマンドによってパケット受信状態になり，クロスバ網からデータを受信して FIFO に格納する．これと同時に PE I/F を介してメッセージサイズを CM へ書き込み，PE に割り込みを掛けて OS にその領域獲得を促す．レシーバは CM から非ゼロの有効なアドレスを得て領域が獲得されたと判断し，PE から LM バスを獲得してメッセージを DMA ライトする．その後，ルータ側で LM バスを解放することにより OS はメッセージのディスパッチを開始する．

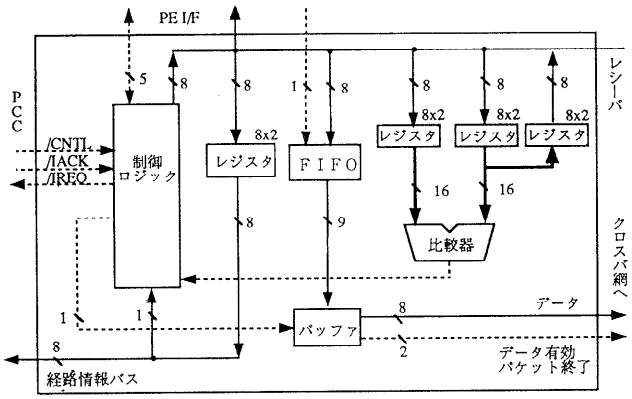


図 4 メッセージセンダ
Fig. 4 Message sender.

メッセージレシーバも，センダとほぼ同じハードウェア構成を持つ．ただし，LM/CM アクセスは PE I/F でシリアライズされるため，CM から読み出すデータの比較に用いる一部のハードウェアは共有している．

5. 性能予測

ここでは，プロトタイプ・ルータにおけるパケット転送速度について考察する．プロトタイプ設計においては，メッセージセンダ／レシーバ，ポートなどの各ブロックのシーケンサを PLD で設計した．また，クロスバ網には TI 社のクロスバスイッチ SN 74 ACT 8841 を，各ブロック内のパケット格納用バッファには IDT 社の FIFO 7200 を用いた．動作周波数は PE と同様な 30 MHz であり，設計に使用したデバイスのスペックから，ルータ間，ルータ内ブロック間共に 2 クロック，0.066 μs で 1 バイトのデータ転送を行う．従って，リンク当りの最大転送能力は 15 MB/s である．

メッセージの送信ノードにおいて，センダが CM からメッセージの宛先部を読み出して PCC に経路選択要求を出すまでに約 40 クロックを要する．PCC のサービス受け付けは，10 クロックで一巡する循環割り当てであるため，他にサービスを要求するブロックがなければ平均 5 クロックでサービスを受け付ける．PCC が経路選択要求を受け付けてから，クロスバ網を設定して送信ポートにそのパケットの転送を通知するまでが 32 クロック，送信ポートにおいて PCC からのコマンドと隣接ノードからの入力の競合がなければ，パケットの先頭がクロスバ網，送信ポートを通ってネットワークに出始めるまで 17 クロックかかる．

したがって，送信ノードから隣接ノードにパケットの先頭が渡されるまでの所要時間は，40+5+32+17=94 クロック，約 3.1 μs である．ヘッダに続くパケットデータは，PCC での経路選択中に次々にセンダ内の FIFO に取り込まれ，ヘッダに引き続いてパイプライン的に送信される．

ポートが隣接ノードからパケット受信を開始すると，ヘッダ部 3 バイトを受信（約 6 クロック）してから PCC に経路選択要求を出す．PCC での経路選択から次の隣接ノードにパケットの先頭が渡されるまでの所要時間は送信ノードと同様である．従って，ネットワークが空いている時，中継ノードで 1 パケットデータ（フリット）を 1 ホップさせるのに，6+5+32+17=60 クロック，約 2.0 μs を要する．隣接ノードからは，2 クロックで 1 バイトのデータが転送されてくるので，1 ルータ内にバッファリングされるのは 60/2=30 バイト程度となり，それ以上のパケットはカットスルー・ルーティングされる．可変長パケットの最大長を 255 バイトとすれば，その時のパケットは 9 (255/30) ノードにまたがって転送されることが分かる．

あるノードに 2 つのパケットが同時に入力された時，一方のパケットは，他方を PCC で経路選択する間の 32 クロック待たされることになるので，1 フリットの 1 ルータ内での所要時間は，60+32=92 クロック，約 3.0 μs となる．重要なのは，この場合であっても各ポートが独自のシーケンサで FIFO にパケットを受け取ることが可能なため，隣接ポートの動作までを止めることがない点である．

また，1 つのルータ内で複数のパケットの送り先ブロック（ポートまたはレシーバ）が衝突した場合に

は，後に経路選択されたパケットはパケットバッファに退避される可能性があるが，平均的なサイズのパケットは 35 バイト程度と小さいため[21]，2度 PCC で経路選択を行う間に送り先ブロックを解放する可能性が大きい．従って，4.1 節で述べたように1度目のブロッキング時にはパケットバッファへ退避しない方がよいと考えられる．

宛先ノードでは，受信ポートがパケットをレシーバに渡し始めるまでに約 60 クロック，レシーバが PE 上の OS にメッセージ書き込み領域獲得の割り込みを発して，実際に LM へ書き込むまでに 80 クロック以上を要する．したがって，宛先ノードでの所要時間は，60+80=140 クロック，4.6 μs 以上となる．

以上のことから，パケットサイズを S，転送距離を D とすれば，無衝突時のパケットの転送遅延時間 (μs) は，

$$T_d = 3.1 + 2.0(D-1) + 4.6 + 0.066 \times S$$
$$= 5.7 + 2.0 \times D + 0.066 \times S$$

で計算できることが分かる．第1項，第2項は，それぞれパケットの先頭が送受信ノード，および中継ノードで消費する時間であり，この和を T_{rt} とする．また，第3項は残りのデータが先頭に続いて到着する時間を表す．S=35，および 255 バイトのパケットが，他のパケットに妨害されず距離 D 転送される時の T_d と T_{rt} を**図5**に示す．

1ホップ当りの遅延時間は，各ポートで決定性のルーティングを行う J-Machine で 60 ns (33 MHz 時)，AP 1000 で 160 ns であるのに対して，A-NET ルータでは約 2.0 μs と大きくなっている．経路選択1回当りの時間が，1フリットの転送（チャネルサイクル）時間と比較してそれほど大きくない場合には，

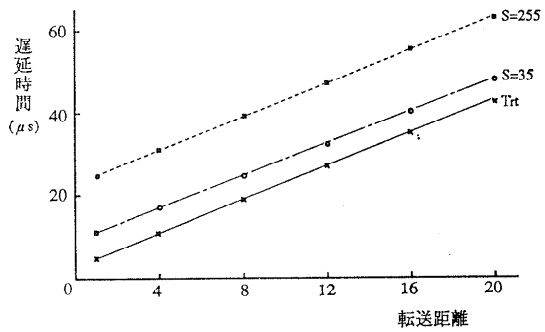


図 5 パケットの遅延時間および経路選択時間と転送距離
Fig. 5 Latency time including routing time of packets vs. transfer distance.

T_d は D に大きく影響されないが[6]，A-NET ルータでは T_{rt} が比較的大きいため，パケットが小さいとストア & フォーワード方式に近くなり，T_d が S と D に大きく影響されることが分かる．これは，トポロジ独立性と適応制御のために，経路選択を PCC で一括管理していることによる．

また，実際にはパケットの衝突やセンダ／レシーバの PE I/F における競合を考慮する必要もあるが，これまでのシミュレーション実験から得た PE 上でのユーザプログラムの1メッセージ当りの平均連続実行時間約 89.2 μs (PE の1マシンサイクル 0.165 μs の時)[1] と比較すると，パケットの転送遅延は1通信当りの内部処理とコンパラブルであるといえる．

6. おわりに

現在，ルータは各ブロックの詳細設計を行っている段階である．ルータ内並列動作を実現するため，各機能ブロックを小規模なステートマシンとして設計している．それぞれのシーケンサは PLD 数個程度で実現され，PCC が全体の動作を調停する．今回報告した方式では，動的オブジェクト生成に伴う DMA 転送をサポートしているため，メッセージ通信だけの場合に比べて制御が複雑になっている．DMA 転送は，あらかじめすべてのノードにクラスを割り付けておけば回避できるため，現在，プログラムのスタートアップやメモリ使用効率などとのトレードオフについても検討している．また，プロトタイプの設計に当たっては，ハードウェア量などを考慮し，動的オブジェクト生成の探索やその応答など本来ルータ間だけでやりとり可能なメッセージも PE で処理することとしたが，これらの役割分担についてもよく検討する必要がある．

PE は現在基板のレイアウト設計を行っており，今後ルータと合わせて2ノードのプロトタイプを試作する予定である．さらに，UNIX ワークステーションの LAN による分散シミュレータを中心とした A-NETL 実行環境を用いて，シミュレーション実験も行っており，今後より詳細なネットワークのスループットや PE 内部処理とのバランスなども調べる予定である．

謝辞 ルータの設計にあたり貴重なご意見を頂きました東芝総合研究所の小柳滋氏に感謝いたします．

なお，本研究は文部省科学研究費補助金，試験研究 B No. 04555077，重点領域研究（超並列処理）No. 04235202，奨励研究 A No. 04750303 による．

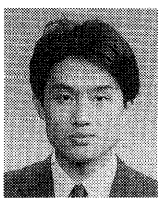
参考文献

1) 馬場敬信，吉永 努：並列オブジェクト指向トータルアーキテクチャA-NETにおける言語とアーキテクチャの統合，信学論 (D-I) 招待論文，Vol. J 75-D-I, No. 8, pp. 563-574 (1992).
2) 馬場敬信：超並列マシンへの道，情報処理，Vol. 32, No. 4, pp. 348-364 (1991).
3) Baba, T. et al.: A Parallel Object-Oriented Total Architecture: A-NET, *Proc. Supercomputing '90*, pp. 278-285 (1990).
4) Baba, T. et al.: A Network-Topology Independent Task Allocation Strategy for Parallel Computers, *Proc. Supercomputing '90*, pp. 878-887 (1990).
5) Dally, W. J. and Wills, D. S.: The J-Machine: A Fine-Grain Concurrent Computer, *Information Processing '89*, Elsevier, pp. 1147-1153 (1989).
6) Dally, W. J.: *A VLSI Architecture for Concurrent Data Structures*, Kluwer Academic Publishers (1987).
7) ダニエル・ヒリス：コネクションマシン，パーソナルメディア，p. 271 (1990).
8) Glass, C. J. and Ni, L. M.: The Turn Model for Adaptive Routing, *Proc. of ISCA '92*, pp. 278-287 (1992).
9) Ishihata, H., Horie, T., Inao, S., Shimizu, T., Kato, S. and Ikesaka, M.: Third Generation Message Passing Computer AP 1000, *Int. Symposium on Supercomputing*, pp. 46-55 (1991).
10) Kermani, P. and Kleinrok, L.: Virtual Cut-Through: A New Computer Communication Switching Technique, *Computer Networks*, Vol. 3, pp. 267-286 (1979).
11) Konstantinidou, S. and Snyder, L.: The Chaos Router: A Practical Application of Randomization in Network Routing, *ACM Computer Architecture News*, Vol. 19, No. 1, pp. 79-88 (1991).
12) Ngai, J. Y. and Seitz, C. L.: A Framework for Adaptive Routing in Multicomputer Networks, *Proc. ACM SPAA '89*, pp. 1-9 (1989).
13) Zorpette, G.: The Power of Parallelism, *IEEE SPECTRUM*, Vol. 29, No. 9, pp. 28-33 (1992).
14) 末吉敏則，杷野公平，有田五次郎：書き換え可能なLSIによる可変構造型相互結合網の実現法，情報処理学会論文誌，Vol. 33, No. 3, pp. 260-269 (1992).
15) 田辺 昇，中村定雄，鈴岡 節，小柳 滋：並列AIマシンProdigyの試作と通信性能評価，信学論 (D)，Vol. J 74-D-I, No. 4, pp. 264-272 (1988).
16) Yalamanchili, S. and Aggarwal, J. K.: Reconfiguration Strategies for Parallel Architecture, *IEEE Computer*, Vol. 18, No. 12, pp. 44-61 (1985).
17) Yantchev, J. T. and Jesshope, C. R.: Adaptive Low Latency, Deadlock-free Packet Routing for Networks of Processors, *IEE Proceedings, Pt. E.*, Vol. 136, No. 3, pp. 178-186 (1989).
18) 米澤明憲：並列オブジェクト指向言語 ABCL/1による並列処理記述とその枠組みの研究，信学論 (D) 招待論文，Vol. J 71-D, No. 8, pp. 1415-1422 (1988).
19) Yoshinaga, T. and Baba, T.: A Parallel Object-Oriented Language A-NETL and Its Programming Environment, *Proc. COMPAC '91*, pp. 459-464 (1991).
20) 吉永 努，鈴木 充，寺岡孝司，茂木 久，馬場敬信：A-NET計算機のノードプロセッサとその実行方式，並列処理シンポジウムJSPP '91, pp. 189-196 (1991).
21) Yoshinaga, T. and Baba, T.: A Local Operating System for the A-NET Parallel Object-Oriented Computer, *J. Inf. Process.*, Vol. 14, No. 4, pp. 414-422 (1992).

(平成4年9月17日受付)
(平成5年3月11日採録)

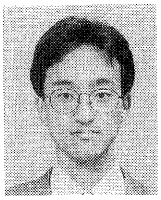
吉永 努 (正会員)

1963年生．1986年宇都宮大学工学部情報工学科卒業．1988年同大学院修士課程修了．同年より宇都宮大学工学部情報工学科助手．並列計算機システム，オブジェクト指向プログラミング言語などに興味を持つ．



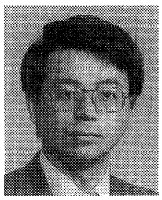
茂木 久 (正会員)

1967年生．1990年宇都宮大学工学部情報工学科卒業．1992年同大学院工学研究科情報工学専攻修士課程修了．同年(株)東芝入社．現在研究開発センター情報・通信システム研究所に所属．並列計算機の研究に従事．



佐々木 昌（学生会員）

1968年生．1991年宇都宮大学工学部情報工学科卒業．1993年同大学院工学研究科情報工学専攻修士課程修了．並列処理システムの諸分野に興味を持つ．



馬場 敬信（正会員）

1947年生．1970年京都大学工学部数理工学科卒業．1975年同大学院博士課程単位取得退学．同年より電通大助手，講師を経て，現在，宇都宮大学工学部情報工学科教授．工学博士．1982年より1年間メリーランド大学客員教授．計算機アーキテクチャ，並列処理，データベースシステムなどの研究に従事．著書「マイクロプログラミング」（昭晃堂），「Microprogrammable Parallel Computer」(MIT Press)，「新しい計算機アーキテクチャ」（丸善，共著）など．電子情報通信学会，IEEE各会員．