ICOT Technical Report: TR-453

TR-453

並列推論マシンPIM/Pのアーキテクチャ

服部　彰、篠木　剛、
久門　耕一(富士通)、後藤　厚宏

March, 1989

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

並列推論マシンＰＩＭ／ｐのアーキテクチャ

服部彰*，篠木剛*，久門耕一*，後藤厚宏**
（* 富士通株式会社，**新世代コンピュータ技術開発機構）

１．はじめに

第５世代コンピュータプロジェクトの一環として，大規模な知識処理プログラムの高速処理をめざして並列推論マシンＰＩＭ／ｐ〔1,2〕を開発中である。

本マシンは並列論理型言語Flat GHC〔３〕をベースとするKL1 言語で記述された大規模知識処理プログラムの高速処理を目的としている。KL1 プログラムはANDストリーム並列に基づく通信を行う多数の微細プロセスから構成される。しかも，この通信は大量データの連続転送ではなく，変数のユニフィケーションのための構造データや負荷分散のためのプロセスの転送であるため，動的であり，頻度が高い。このため，プロセス間の通信コストを下げることが本マシン高速化のための重要な設計方針である。

このためには，数100 台の高速プロセッサをメモリ共有の密結合にすることが理想であるが，メモリや結合部のスループットがプロセッサ台数に比例した並列処理性能実現のネックとなる。そこで，10台程度のプロセッサをメモリ共有密結合したクラスタをネットワークにより結合する二階層構成を採った。このような二階層システムを高速化するためには，クラスタ内のメモリ・バス系のトラフィックを減らさねばならない。また，ソフトウェアの通信の局所性を有効に引き出すために，クラスタ間ネットワークは高速であると共に柔軟にしてソフトウェアから使い易くせねばならない。

本稿では，以上のような観点から要素プロセッサとクラスタそしてネットワークに採用した高速化アーキテクチャとその予備評価について報告する。

２．システム構成

2.1 クラスタによる二階層接続

KL1 言語では共有変数やゴール（プロセス）の排他的アクセスを高速に行うために同期機能を持った共有バスによる接続が不可欠である。従って，図１に示すように共有バスの性能が許す範囲から８台の高速プロセッサをメモリ共有バス結合にしてクラスタを構成した。クラスタ内では一つのアドレス空間を共有し，高速なプロセス間通信による並列処理が可能である。

各クラスタは数１００プロセッサ規模への拡張性と高い並列処理性能のために，図２のようにネットワーク結合されている。クラスタ内のプロセッサ４台づつが１つのルータを共有して，ハイパーキューブのノードとして結合されている。このネットワークはKL1 実行時の分散ユニフィケーションとゴールの負荷分散のための通信路を提供し，非同期メッセージのパケット通信が行われる。

2.2 入出力インタフェース

本マシンはワークステーションをフロントエンド(FEP) とするバックエンドマシンである。ユーザインタフェースや基本的なファイル入出力はFEP に任せる。しかし，高い推論性能に見合った入出力処理を可能とするため，複数の入出力機器，特にディスク装置を本体に直結する必要がある。そこで，各クラスタの複数(6台) プロセッサに入出力インタフェース(SCSI)を持たせた。

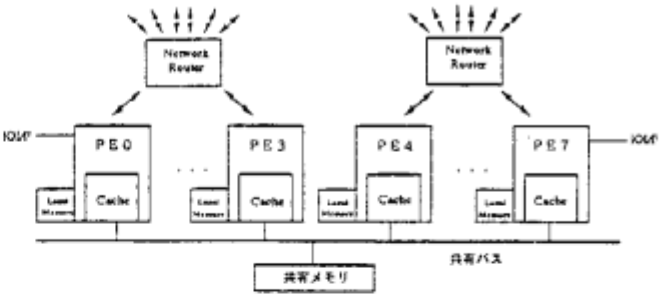

図１　クラスタの構成

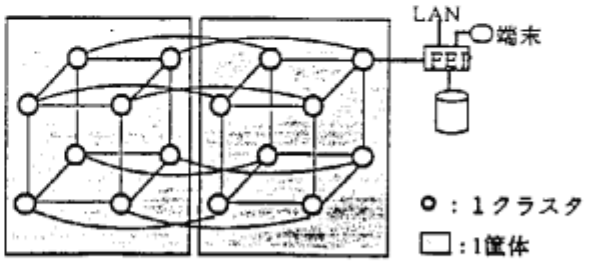

図２　システム構成

3．要素プロセッサ

3.1 設計方針

KL1 言語の並列処理を効率良く行うために，次の点に重点を置いて要素プロセッサの設計を行った。

(1)知識処理向きアーキテクチャ

KL1 プログラムでは実行時のデータ型チェックが多用されるため，タグアーキテクチャを導入した。タグ判定命令やタグ付与命令等のタグ付き語を操作する命令を備えた。

KL1 プログラムは単一代入やプロセス間のAND ストリーム通信のためにメモリを大量消費し，回収処理のコストが大きい。従って，実時間ガーベジコレクション(MRB方式〔4〕)の高速化のため，タグ中のMRB フラグを扱う機構を備えている。

(2)RISC型パイプラインアーキテクチャ

高集積LSI を活用してマシン命令実行速度を高めるため，RISC型マシン命令のパイプライン処理を採用した。即ち，演算とメモリアクセスを分離したRISC型命令によりパイプラインを簡単化して命令実行部を1チップ化してマシンサイクルの短縮を図った。

(3)マルチプロセッサ向きアーキテクチャ

共有バスを介したメモリ共有結合を採るクラスタの性能を高めるためには，バストラフィックを低減することが不可欠である。そのため各プロセッサはライトバック型キャッシュメモリを持つ。更にバス上を流れるプログラム量を減らすために命令を高機能化するための機能（マクロ命令）を持つ。また，バス上を流れるデータ量を減らすためにキャッシュメモリの一部の動作を制御するためのキャッシュ制御命令を備えた。

3.2 命令実行方式の選択

KL1 言語は実行時の動的なデータ型判定が必要であり，データ型により処理内容が多岐に分かれる。このようなPolymorphic な機能を実行する方式として次の二つの方式が考えられる。

(1)KL1 プログラムを高機能なマシン命令列にコンパイルして，それをマイクロプログラムによりエミュレートする方式

(2)KL1 プログラムをRISC的なマシン命令列にコンパイルしてハードウェアで直接実行する方式

RISC的なマシン命令は短いパイプラインと短いマシンサイクルで実行できる。またコンパイラによる最適化が可能である。しかし，静的コードサイズが大変大きくなる。これはキャッシュメモリのヒット率を低下させて共有バスのトラフィックを増加させる。

一方，高機能マシン命令の場合には静的コードサイズを小さくできるため，キャッシュメモリのヒット率が向上して共有バスのトラフィックが低減する。しかし，コンパイラによるコード最適化の余地が狭められる。また，マイクロムログラムによるエミュレートでは制御記憶へのディスパッチ時間のためマシンサイクルが長くなる。

そこで，両方式の長所を合わせた方式（マクロ命令メカニズム）を開発した。

即ち，RISC 的マシン命令に加えて，オペランドデータの型判定に基づく条件付きサブルーチンコール命令（マクロ呼出命令）を用意した。主記憶に格納されたプログラムはRISC的命令とマクロ呼出命令から成る。

3.3 マクロ命令メカニズム〔5，6〕

要素プロセッサは外部命令と内部命令の2種類のマシン命令を持つ。外部命令は主にユーザプログラムのコンパイルコードを表現するのに使う。外部命令は高機能（マクロ）命令を構成する内部命令ルーチンを呼び出すためのマクロ呼出命令を含む。マクロ呼出命令は最初にオペランドとして与えられたレジスタ上のデータ型をテストし，その結果によりマクロ本体を呼び出す。マクロ本体は内部命令で記述されプロセッサ内部のメモリ（内部命令メモリ）に格納されている。他どの外部命令と内部命令は共通のRISC的命令であるため，共通のパイプラインを使うことができる。

マクロ呼出命令は次の形式を持つ。

MacroCall　<cond> <address> <R1> <Tagimm/R2>

<address> ：マクロ本体の内部命令メモリアドレス

<cond>　　：And , Or , Xor ・・・

機能は，<R1> で指定されたレジスタのタグ部と即値<Tagimm>または<R2>で指定されるレジスタのタグ部を<cond>の条件で比較して条件が成立すれば<address>で指定されるマクロをマクロ呼出する。条件成立時，レジスタ番号<R1>，即値<Tagimm>またはレジスタ番号<R2>が間接レジスタに設定される。これらはマクロ本体からオペランドを間接指定でアクセスする時に使われる。

図3に要素プロセッサのマクロ呼出機構とパイプラインの関係を示す。主記憶にはRISC的命令とマクロ呼出命令が混在したコンパイルコードが格納されている。キャッシュメモリを通って命令バッファに読み出された外部命令の内，マクロ呼出命令以外の命令はそのままパイプラインに投入されて実行される。マクロ呼出命令の場合にはディスパッチ機構が働き，内部命令メモリにあるマクロ本体の内部命令列の実行に制御が移る。マクロの最後の内部命令が実行されると，命令バッファに来ている次の外部命令の実行に制御が帰る。

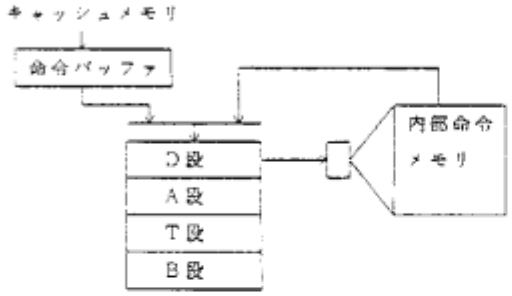

図3　マクロ呼出機構

3.4　パイプライン構造

要素プロセッサは，ＤＡＴＢの4段のパイプラインを基本とする動作を行い，基本的にマシンサイクル毎に1命令づつ実行出来る。

演算命令はレジスタ上のオペランドのみを対象としメモリアクセス命令はレジスタとメモリの間の転送だけで演算は行わない。これにより，パイプライン段数を2段短縮した。

内部命令は上記の4段に加えて，D段の前に内部命令メモリのアドレス設定（S段）と内部命令メモリ読み出し（C段）のために2段必要である。

条件成立の場合

```
D  A                          ：マクロ呼出命令
   D                          ：抑止された外部命令
S  C  D  A  T  B              ：最初の内部命令
   S  C  D  A  T  B           ：2番目の内部命令
```

条件不成立の場合

```
D  A                          ：マクロ呼出命令
   D  A  T  B                 ：次の外部命令
      D  A  T  B              ：外部命令
```

マクロ本体の終了

```
S  C  D  A  T  B              ：EOI 付き内部命令＊
   S  C                       ：抑止された内部命令
      S                       ：抑止された内部命令
         D  A  T  B           ：次の外部命令
```

＊ EOI：End Of Instruction

図4　マクロ呼出命令のパイプライン動作

表1　パイプラインの構造

段	演算命令	メモリアクセス命令
D	Decode	Decode／reg. read
A	—	Address calculation
T	Register read	Cache access(Tag)
B	ALU/ reg. write	Cache access／reg. write

外部命令の条件分岐では条件判定をB段で行っているが，マクロ呼出命令の場合には内部命令メモリの読み出しタイミングに合わせてA段で条件判定を行っている。マクロ呼出命令は分岐ペナルティが2サイクル（図4参照）で，外部命令の分岐に比べて2サイクル短い。また，マクロ本体からの復帰ペナルティは0で外部命令のサブルーチンからの復帰に比べて4サイクル短い。

3.5　ライトバック方式キャッシュメモリの導入

KL1言語は単一代入の性質から，従来言語に比べてメモリ書込みが多い。従って，共有バスのトラフィックを減らすためにライトバック方式の一種であるIllinois方式［7］を基本とするキャッシュメモリを採用した。

(1)ブロックサイズ

KL1言語は高頻度のプロセス間通信を行うため，ブロックサイズを大きくし過ぎると，キャッシュ間の無効化が増加し，バストラフィックの増加が大ブロックの先取り効果を消してしまう。また，タグ(Address Array)の物理的サイズと必要なキャッシュ容量の確保も考慮して32 byte（4タグ付き語）ブロックを採用した。

(2)構成

命令用とデータ用にそれぞれ64 KByte容量を持つが，内容は同一である。構成は，それぞれ328 ブロック×2セクタ×256 カラム×4セットである。

(3)キャッシュプロトコル

Illinois方式を基本とし，これにロック機構を実現するための状態と3.6節で述べるキャッシュ制御命令を追加した。各32Byteブロック毎に次の6状態の内のいずれかを持つ。

ＥＭ	Exclusive Modified
ＥＣ	Exclusive Clean
ＳＣ	Shared Clean
Ｉ	Invalid
ＥＭＬ	Exclusive Modified with Lock
ＥＣＬ	Exclusive Clean with Lock

3.6　キャッシュ制御命令の導入

KL1言語の特性から，コンパイラがメモリのある領域や語のデータが有効でないことを認識できる場合が

ある。そこで，共有バス上の無駄なデータ転送を抑制するためにキャッシュの動作を制御する命令を追加した。主な命令について以下に述べる。

(1) Direct Write 命令

KL1 言語ではヒープからの未使用の領域の切り出しや，生成したゴールの制御ブロックの作成が頻繁に行われる。このような初めて使う領域への書き込みの際には，キャッシュミスヒットの場合でも主記憶からの読込を省いて，キャッシュ上に確保したブロックに直接書き込みを行うことが可能である。

そこで，上記機能を持った本命令を導入して不要な主記憶からの読込の発生を防ぐ。

(2) Read Invalidate命令

プロセッサ間のメッセージ転送やゴールの分散の場合，受信側が受け取った後は送信側がそれらのデータを参照する可能性は低い。送信プロセッサのキャッシュに残っていると受信プロセッサが書き換えた場合に無駄なバッファ無効化コマンドが送信される。

この命令は読込操作と共に送信側のキャッシュブロックをついでに無効化することにより，上記の無駄なバッファ無効化コマンドの発生を防ぐ。

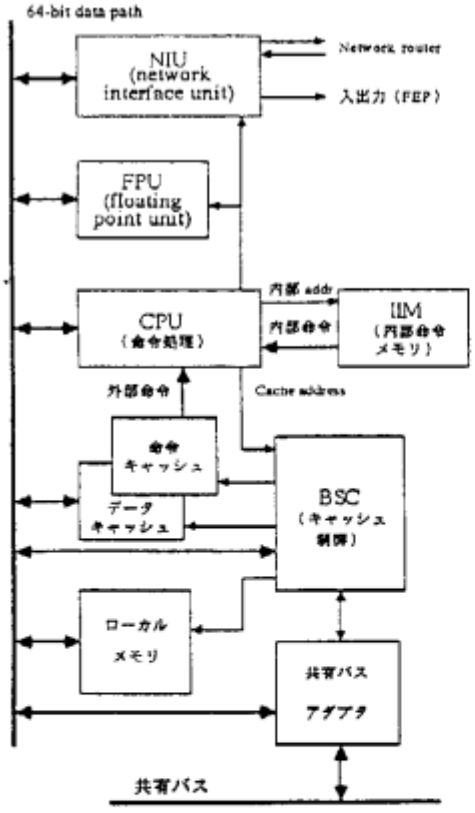

図5　要素プロセッサの構成

4．クラスタ性能の評価

本章ではシミュレーションに基づいて，本マシンの性能の基本となるクラスタ性能を算定評価する。

4.1　クラスタ性能の要点

図1に示したように，8台の要素プロセッサが共有バスを介して主記憶（最大256MB)を共有している。共有バスの転送速度はデータ幅8 Byte，バスサイクル50 ns(目標) である。

プロセッサ台数に比例した高いクラスタ性能を実現するには，共有バスのトラフィックを減らすことが最重要である。そのために，大容量のライトバック型キャッシュメモリを基本として，マクロ命令メカニズムやキャッシュ制御命令がどれだけバストラフィックの低減に効果があるかがポイントとなる。

4.2　マクロ命令によるコードサイズ縮小の予測

KL1 コンパイラのマシン命令レベルのコード生成部は未だ出来ていない。そこで，マクロ命令によるコードサイズの縮小(無いことによる増加) 率を予測するために，小さなベンチマークプログラムをハンドコンパイルして算定を行った(表2)。

動的ステップ数の増加は静的ステップ数に比べてかなり少ない。しかし，キャッシュメモリを備えたプロセッサの場合にはブロック単位に読み込まれるため，アクセスされた近傍のコードもキャッシュメモリに入る。従って，共有バス負荷としての動的ステップ数は静的ステップ数にかなり近づくと予想される。

表2　マクロ命令によるステップ数の変化

	マクロ	静的step数比	動的step数比
append	有	1	1
	無	3.5	1.2
qsort	有	1	1
	無	3.5	1.3

4.3　シミュレーション

2つのプログラムについてキャッシュのソフトウェアシミュレータにより共有バスのトラフィックに関する測定を行った。

(1)シミュレータの測定条件

プロセッサ台数は8台である。キャッシュメモリの構成は，4語ブロック，2セクタ，256 カラム，4セットである。従って各プロセッサ毎に8 K語である。

(2)各プログラムの条件

①標準：Direct Write命令有り，Read Invalidate 命令無，KL1-b コードを実行した。

②コードサイズ2倍：KL1-b コードの静的，動的サイズを2倍に拡大して実行した。

③コードサイズ4倍：KL1-b コードの静的，動的サイズを4倍に拡大して実行した。

④Direct Write命令無：Direct Write命令を普通のWrite 命令にし，他は標準と同じ。

⑤RI変更：ヒープ領域へのRead命令をRead Invalidate 命令に変更して実行，他は標準と同じ。

(3)測定結果

表3に示すように，コードサイズが2倍，4倍となると，バスサイクルが各々平均で約10%，70%増加している。また，Direct Write命令が無い場合にはバスサイクルが平均で約30%増加している。これらはマクロ命令やDirect Write命令のバスサイクルを減らす効果が大きいことを示す。

Read Invalidate 命令に変更した測定ではバスサイクルが平均で5%程しか減っていない。これはヒープ領域への全てのRead命令をRead Invalidate 命令に変更したためであり，選択的に変更した場合には更にバスサイクルが減少することが期待される。

尚，各プログラムのリダクション数は次の通りであった。

BUPのリダクション数　：117,334

8Queen のリダクション数：116,635

(4)クラスタ性能の算定

表3のBUP プログラムのデータに基づいてクラスタ性能を算定した結果を表4に示す。但しプロセッサの単体性能を400KLIPSと仮定している。

表5に示すように，コードサイズが2倍，4倍となるとクラスタ性能が各々約5%，30%低下する。また，Direct Write命令が無い場合にはクラスタ性能が約6%低下する。

表4のクラスタ性能は次のように算定した。

Tpe／(Tpe + Tcb + Tw)

ここでTpe はプロセッサ当たりのリダクション時間（約 117K サイクル）である。

プロセッサ当たりの共有バス使用サイクルを全て性能低下要因にしているのは，バスサイクルの80%以上がキャッシュ間転送であるためこの時間はプロセッサは動作出来ないと見なせるからである。

表3 シミュレーションによる共有バスサイクル数
（単位：1000）

	標準	codesize 2倍	codesize 4倍	DW無し	RI変更
BUP	512	643	1217	654	507
標準比	1	1.26	2.38	1.28	0.99
8Queen	337	344	356	461	306
標準比	1	1.02	1.06	1.37	0.91
平均	1	1.14	1.72	1.32	0.95

表4　クラスタ性能の算定（BUPの場合）

（Tcb と Tw の単位：1000 サイクル）

	標準	code size 2倍	code size 4倍	DW無し	RI変更
名目バス使用率	70%	88%	166%	89%	69%
PE当りバス使用サイクル（Tcb）	64	80	152	82	63
PE待時間（Tw）	5.5	10.2	57.8	10.5	5.4
クラスタ性能比	1	0.95	0.67	0.94	1.0

4．クラスタ間ネットワーク

本節では，クラスタ間ネットワークを設計する際に考慮した要点と，それに基づき採用したアーキテクチャについて述べる。

4.1 設計方針

このネットワークは数100 台のプロセッサを結合し，その上をKL1 言語の分散ユニフィケーションやゴール分散のための比較的短い可変長データが動的に転送される。ネットワークのトポロジーとして，その直径が短く，ハードウェア量の割に総スループットが高く，そして分散制御が可能な点からハイパーキューブを採用した。

(1)充分なスループットの確保

クラスタからの通信要求を充分に処理できるスループットを確保する。

(2)デッドロックフリー

KL1 言語はプロセス間の動的で柔軟な通信が必要なため，パケット交換ネットワークを構成する上で，デッドロックが最も重大な問題の一つとなる。特に，Store and Forward 型デッドロック〔8〕を防ぐように設計せねばならない。

(3)動的負荷分散のサポート

負荷分散は，一般にまず静的負荷分散により概略のプログラムやデータの配置を行う。しかし知識処理プログラムの動作をコンパイル時に完全にスケジュールすることは難しいため，動的負荷分散による調整が必要である。

(4)ハイパーキューブのスループットの有効利用

ハイパーキューブのノードに対する入出力トラフィックはノード間トラフィックの2倍まで可能である。従って，複数プロセッサをネットワークノードに接続して共用すべきである。これはハイパーキューブの次数を下げて，ネットワークの直径やクラスタ間の信号線数を縮小する点でも有利である。

(5)プロセッサとネットワークを直結

パケットの送受信を共有バスを通さずに，コプロセッサインタフェースとしてプロセッサとネットワーク

ノードを直結させる。これにより，共有バスのトラフィックが削減できる。

4.2 ネットワークの基本構造と性能見積

図1に示すように，4台の要素プロセッサがネットワークルータ・ノードに接続されている。各クラスタは2個のルータを持っており，二重のハイパーキューブを構成している。各ルータは6本のクラスタ間リンクを持つ。つまり最大64クラスタから成る6次のハイパーキューブを構成できる。

次に，ネットワークに要求されるスループット性能と余裕度について述べる。要素プロセッサの推論速度を400KLIPSとし，10リダクションに1回の割合で 100byteの通信を行うと仮定すると，各ルータノードには16MB/Sの通信要求が来る。通信先がランダムとするとクラスタ間リンクのトラフィックは8MB/Sとなる。後述するように，各リンクの転送速度は20MB/Sであるから，この場合の負荷率は40%と見積もられる。

4.3 デッドロックフリーなルーティング方式

パケット交換ネットワークではStore and Forward型のデッドロックが本質的で重要である。ハイパーキューブのこの型のデッドロック回避法としては，ルーティングに依存しない方法(Structured Buffer Pool法〔9〕，以下SBP 法と略す）とルーティングを制限する方法（最下位ビット優先ルーティング法または E-cube〔10〕と呼ばれる）がある。

SBP 法は混んでいるルートを動的に回避することが可能でスループットが向上する反面，パケットの順序が保証されないこととルータのバッファ制御のハードウェアが複雑になる欠点がある。

最下位ビット優先法は経路が一意的となるが，パケットの順序が保証されることとバッファは簡単なFIFOで済む。 そこでシミュレーションにより両方式のスループットを比較した。

(1)シミュレーションの基本条件

4次元のハイパーキューブ（16ノード）を使い，各ノードにはパケットバッファを256byte ×4個持たせた。パケット長は0～Nの一様乱数とし，最大値Nは100 ～5000byteについて調べた。転送の要求源であるプロセッサは各ルータノードに4台接続した場合と1台接続した場合について調べた。

(2)バッファバンクの選択方法

SBP 法ではノード間でパケットを転送する時，送信ノードがバッファバンクn を使った時には受信ノードは n+1 以上のバンクを選択する必要がある。バンクを逐次に選択する場合と使用バンクの偏りを減らすために飛び越して選択する場合についても調べた。

(3)シミュレーション結果

プロセッサからルータノードへの転送要求（ 10MB/S, 20MB/S, 40MB/S ）に対して実際に送出された割合を，平均パケット長1000byteの場合について図6に示す。横軸は次のようなルーティング方法を示す。

det ：最下位ビット優先法

str.r.cap ：SBP 法（飛び越しバンク選択）

str.r ：SBP 法（逐次バンク選択）

図6から分かるように，固定ルーティングである最下位ビット優先法のスループットは可変ルーティングであるSBP 法と大差がないことが分かった。従って，クラスタ間ネットワークのルーティングには最下位ビット優先法を採用した。

4.4 ハイパーキューブのスループットの有効利用

ルータノードに複数プロセッサを結合することにより，送信先がよりランダマイズされるためハイパーキューブのスループットを有効利用できる。図6の結果から，接続プロセッサが4台の方が1台に比べて多く転送できることが分かる。

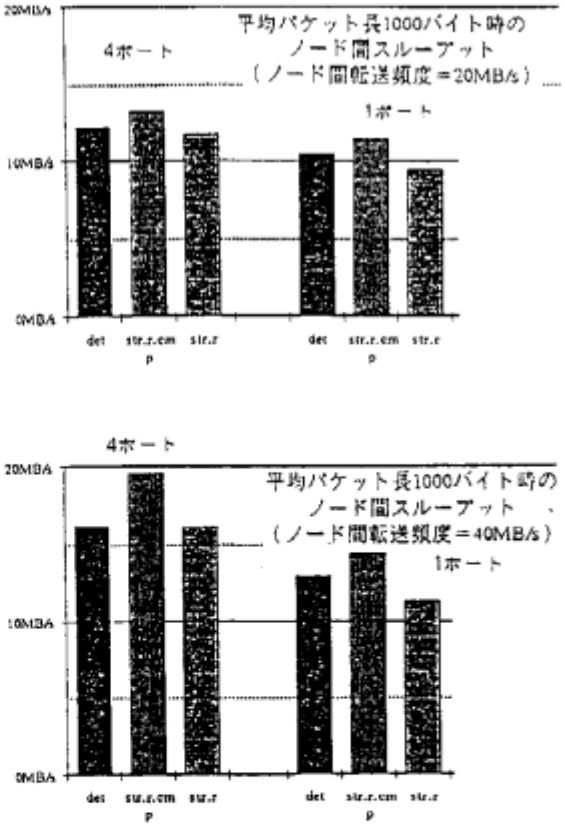

図6 ネットワークスループットとルーティング方式

4.5 ネットワークの構造

ネットワークルータの内部構造を図7に示す。ルータは基本的に11入力（クラスタ：6，プロセッサ：4，負荷分散機構等の制御部：1）×11出力のクロスバスイッチである。クラスタの6ポートの内，2ポートはクロック同期転送用であり同一筐体内の近接クラスタとの接続に使う。残り4ポートは非同期転送用で離れたクラスタとの接続に使う。

表5　クラスタ間ネットワークの諸元

転送速度	：20MB/S(/リンク)
クロック周波数	：50ns
パケット長	：8KB（最大）
デッドロックフリー機能	：固定ルーティング
動的負荷分散サポート機構	

5．まとめ

現在開発中の並列推論マシンPIM/p の高速化アーキテクチャの特徴と，特に次の点を述べた。

(1)クラスタの共有バストラフィックを低減するために要素プロセッサに採用したマクロ命令機構とキャッシュ制御命令によりクラスタ性能をかなり向上できる見込みである。

(2)クラスタ間ネットワークのデッドロックフリーなルーティング方式として固定ルーティング（最下位ビット優先方式）で充分なスループットが得られることをシミュレーションにより確認し採用した。

(3)クラスタ間ネットワークの各ルータノードに複数プロセッサを直接結合することにより，ハイパーキューブのスループットを有効利用できることをシミュレーションにより確認し採用した。

謝辞

日頃御指導頂く富士通研究所機構情報処理研究部門長，林人工知能研究部長ならびにＩＣＯＴ内田第４研究室長に感謝致します。また，クラスタバス負荷のシミュレーションデータを測定された三菱電気松本明氏および富士通，ＩＣＯＴのPIM/p 研究開発メンバ諸氏に感謝致します。

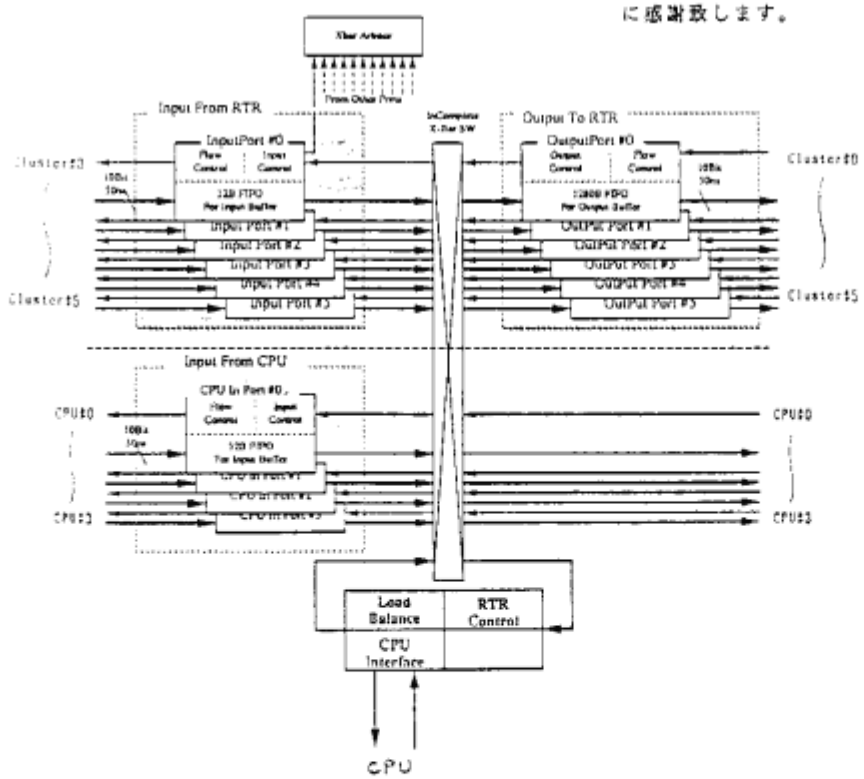

図7　ネットワークルータの構成

参考文献

〔1〕後藤他：並列推論マシンPIM/p の概要，情報処理学会第37回全国大会予稿集pp135，昭和63年

〔2〕A.Goto他：Overview of Parallel Inference Machine Architecture(PIM)，Int. Conf. on FGCS 1988，pp208.

〔3〕上田：Introduction to guarded horn clauses TR209，ICOT，1986.

〔4〕近山他：Multiple Reference Management in Flat GHC，In Proc. of the 4th ICLP,(1987)

〔5〕榎木他：並列推論マシンPIM/p の要素プロセッサのアーキテクチャ，情報処理学会第37回全国大会予稿集pp137，昭和63年

〔6〕T.Shinogi 他：Macro-call Instruction for the Efficient KL1 Implementation on PIM，Int. Conf. on FGCS1988，pp953.,(1988)

〔7〕J.Archibald 他：An Evaluation of Cache Coherence Solution in Shared-Bus Multiprocessors, TR 85-10-05，Dep. of Comp. Science，Univ. of Washington.

〔8〕P.Merlin他:Deadlock Avoidance in Stored-and-Forward Networks-Ⅱ:Stored-and-Forward Deadlock，IEEE Trans. Comm. Vol.28，No.3，pp345.(1980)

〔9〕E.Raubold 他：A Method of deadlock-Free Resource Allocation and Flow Controll in Packet Networks，Proc. ICCC1976，pp483.(1976)

〔10〕W.Daily 他：Deadlock-Free Message Routing in Multiprocessor Interconnection Networks, IEEETrans. Comp.，Vol. C-36，No.5.(1987)