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†††

本マシンは並列論理型言語 KL 1 で記述された知識処理プログラムの高速処理を目的とする数百台規模のマルチプロセッサであり，第五世代コンピュータプロジェクトの中で開発中である．KL 1 言語で多用されるプロセス間の通信・同期を高速処理するため，10 台程度のプロセッサ (PE) をメモリ共有バス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．本論文では特に，クラスタとネットワークで採用した高速化アーキテクチャの主な特徴と評価について述べる．PE 台数に比例したクラスタ性能を実現するためには，共有バスの通信量を減らすことが重要である．共有バス上の命令コード量とオペランドデータ量を減らすために，命令を動的に高機能化するマクロ命令呼び出し機構とキャッシュメモリの動作をソフトから制御する命令をそれぞれ設計した．これらの機構や命令がバスの通信量の低減とクラスタ性能の向上に有効なことをシミュレーションにより確認した．次に，クラスタ間のハイパキューブネットワークの各ルータノードに複数 PE を接続することにより，ネットワーク性能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．また，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法として固定ルーティング法でも十分なスループットが得られることを確認し，採用した．

1. はじめに

第五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 KL 1 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．KL 1 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，数百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ともに柔軟にしてソフトウェアから使いやすくせねばならない．

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

KL 1 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．したがって，**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．

各クラスタは数百プロセッサ規模への拡張性とプロセッサ台数に比例した並列処理性能を実現するために，**図 2** のようにネットワークにより結合されている．クラスタ内のプロセッサ 4 台ずつが一つのルータを共有して，ハイパ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 KL 1 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．

† The Architecture of Parallel Inference Machine PIM/p by Akira Hattori, Tsuyoshi Shinogi, Kouichi Kumon (Artificial Intelligence Laboratory, FUJITSU LABORATORIES LTD.) and Atsuhiro Goto (Fourth Research Laboratory, Institute for New Generation Computer Technology).

†† (株)富士通研究所人工知能研究部
††† (財)新世代コンピュータ技術開発機構第 4 研究室

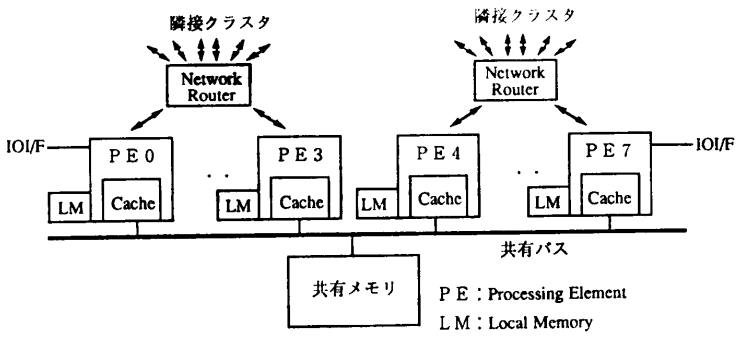

図 1 クラスタの構成
Fig. 1 Configuration of a cluster.

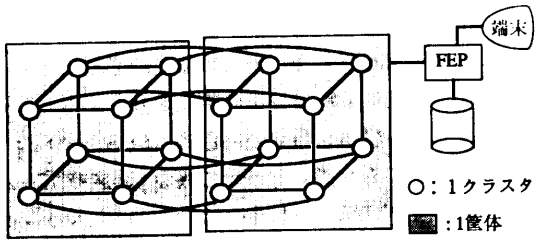

図 2 PIM/p システム構成
Fig. 2 Configuration of PIM/p system.

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．そこで図1のように，各クラスタの2台のプロセッサに入出力インタフェース (SCSI) を持たせた．

3. 要素プロセッサ

3.1 設計方針

KL 1 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

KL 1 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タグアーキテクチャを導入した．

KL 1 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．したがって，実時間ガーベジコレクション (MRB 方式[4]) の高速化のために，タグ中の MRB フラグを扱う機構を備えている．

(2) 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．さら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

KL 1 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．このような機能を効率良く実現する方式として次の二つの方式が考えられる．

(1) KL 1 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) KL 1 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[5] 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減少する．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，マイクロプ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に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．

すなわち，3.4 節で述べるように 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（マクロ呼び出し命令）を用意した．主記憶に格納されたプログラムは RISC 的命令とマクロ呼び出し命令から成る．

3.3 マクロ命令メカニズム[6),7)]

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．外部命令は主記憶に格納され，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び出し命令を含む．マクロ呼び出し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の型をテストし，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ており，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．ほとん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で実行することができる．

マクロ呼び出し命令は次の形式を持つ．

MacroCall〈cond〉〈address〉〈R 1〉〈Tagimm/R 2〉

〈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，〈R 1〉で指定されたレジスタのタグ部と即値〈Tagimm〉または〈R 2〉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呼び出す．マクロ呼び出し時にはレジスタ番号〈R 1〉，即値〈Tagimm〉またはレジスタ番号〈R 2〉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レジスタ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び出し機構とパイプラインの関係を示す．主記憶には RISC 的命令とマクロ呼び出し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うち，マクロ呼び出し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．マクロ呼び出し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，命令バッファにきている次の外部命令の実行に制御が帰る．

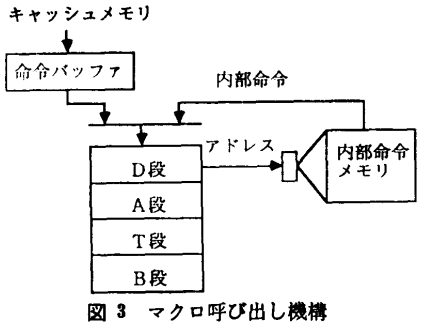

図 3 マクロ呼び出し機構
Fig. 3 Macro-call mechanism.

3.4 パイプライン構造

要素プロセッサは，**表 1** のように DATB の 4 段のパイプライン動作を行い，全命令が 1 マシンサイクルごとにパイプライン処理される．また，演算命令はレジスタ上のオペランドのみを対象とし，メモリ参照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．つまり，演算とメモリ参照を分離してパイプラインを簡単化し，命令実行をハード布線制御にしている．以上の点から，マシン命令数は 190 個と豊富であるが，命令の基本タイプと実行方式は RISC 的[5]である．

内部命令の実行時には上記の 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．

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，マクロ呼び出し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．マクロ呼び出し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．

```
条件成立の場合
  D  A                          ：マクロ呼び出し命令
     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                          ：マクロ呼び出し命令
     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 マクロ呼び出し命令のパイプライン動作
Fig. 4 Pipeline operation of a macro-call instruction.

表 1 パイプラインの構造
Table 1 Structure of pipeline.

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

3.5 ライトバック方式キャッシュメモリの導入

KL 1 言語は単一代入の性質から，従来言語に比べてメモリ書き込みが多い．したがって，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[8]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

KL 1 言語は高頻度のプロセス間通信を行う．ブロックサイズを大きくし過ぎると，プロセッサ間のデータ共有のためにキャッシュ間の無効化コマンドが増加し，バストラフィックの増加がキャッシュの先取り効果を消してしまう．また，キャッシュタグ (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 (図 5 参照)．構成は，それぞれ 32 Byte 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を追加した．各 32 Byte ブロックごとに次の 6 状態のうちのいずれかを持つ．

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

KL 1 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

KL 1 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み込みを省いて，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み込みの発生を防ぐ．

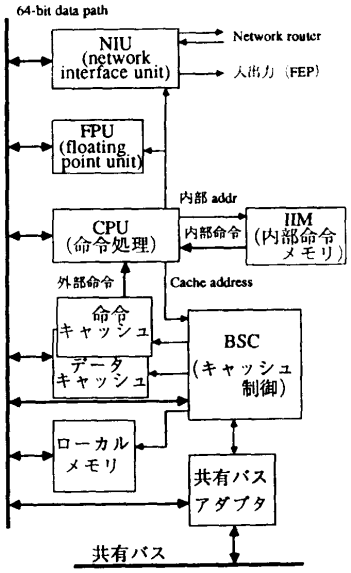

図 5 要素プロセッサの構成

Fig. 5 Configuration of a processing element.

（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．

この命令は読み込み操作とともに送信側のキャッシュブロックを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．

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 256 MB）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も重要である．そこで，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を評価する．

4.2 マクロ命令によるコードサイズ縮小の予測

KL 1 コンパイラのマシン命令レベルのコード生成

表 2 マクロ命令によるコードサイズの変化
Table 2 Effect of macro-instructions on program code size.

	マクロ	静的コードサイズ比	動的コードサイズ比
append	有	1	1
	無	3.5	1.2
qsort	有	1	1
	無	3.5	1.3

部はまだできていない．そこで，マクロ命令を用いることによるコードサイズの縮小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** 参照）．

なお，表 2 の動的コードサイズ比は概ね外部命令ステップ数比に対応し，マクロ呼び出し時間やマクロ本体の実行時間は含んでいないので実行時間比ではない．

動的コードサイズの縮小は静的コードサイズ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．したがって，共有バス負荷としての動的コードサイズは静的コードサイズにかなり近づくと予想される．

4.3 シミュレーション

二つのプログラムの KL 1-b コード[9]をエミュレータ上で実行し，命令フェッチのアクセスパターンをキャッシュのシミュレータにより解釈して，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．したがって各プロセッサごとに 8K 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 命令無しの KL 1-b コード[9]を実行した．

② コードサイズ 2 倍：KL 1-b コードの各命令サイズを 2 倍に拡大して実行した．

③ コードサイズ 4 倍：KL 1-b コードの各命令サイズを 4 倍に拡大して実行した．

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

⑤ RI 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，バスサイクル数が各々平均で約 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 命令を Read Invalidate 命令に変更した測定ではバスサイクル数が平均で 5% ほどしか減っていない．これはヒープ領域へのすべての 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さらにバスサイクルが減少することが期待される．

なお，各プログラムのリダクション数は次のとおりであった．

BUP のリダクション数 ：117,334

8 Queen 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．ただしプロセッサの単体性能を 400 KLIPS と仮定している．

表 4 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

表 3 共有バスの使用サイクル数
Table 3 The number of used common-bus cycles.
〔単位：1,000 サイクル〕

	標準	codesize 2 倍	codesize 4 倍	DW 無し	RI 変更
BUP	512	643	1,217	654	507
標準比	1	1.26	2.38	1.28	0.99
8 Queen	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)
Table 4 Evaluation of cluster performance (in case of BUP).
〔Tcb と Tw の単位：1,000 サイクル〕

	標準	codesize 2 倍	codesize 4 倍	DW 無し	RI 変更
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

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のクラスタ性能比は次のように算定した．

$$Tpe/(Tpe+Tcb+Tw)$$

ここで Tpe はプロセッサ当りのリダクション時間（約 733 K サイクル，1サイクル＝50 ns）である．

プロセッサ当りの共有バス使用サイクルをすべ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できないと見なせるからである．

5. クラスタ間ネットワーク

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

5.1 設計方針

このネットワークは数百台のプロセッサを結合し，その上を KL 1 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キューブを採用した．

（1） 十分なスループットの確保

クラスタからの通信要求を十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

KL 1 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[10]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実行時の調整が必要である．

（4） ハイパキューブのスループットの有効利用

ハイパ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．したがって，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本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（図5参照）．これにより，共有バスのトラフィックが削減できる．

5.2 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 400 KLIPS とし，10 リダクションに1回の割合で 100 Byte の通信を行うと仮定すると，各ルータノードには 16 MB/s の通信要求がくる．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 8 MB/s となる．後述するように，各リンクの転送速度は 20 MB/s であるから，この場合の負荷率は 40% と見積もられる．

5.3 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[11]，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[12] 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，パケットの到着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径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 256 Byte×4 個持たせた．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．KL 1 の分散メモリでの通信の特性については，まだ知られていないため，送信先はランダ

ム，パケット長は 0～N の一様乱数とし，最大値 N は 200 Byte から 5,000 Byte の場合について測定した．

（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 番を使った時には受信ノードは $n+1$ 番以上のバンクを選択する必要がある．受信ノードのバンクとして必ず $n+1$ 番を選択する場合（逐次選択）と，使用バンクの偏りを減らすために $n+1$ 番以上で空いているバンクを選択する場合（飛び越し選択）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（20 MB/s）に対して実際に得られた転送速度を**表 5** に示す．横軸は次のようなルーティング方法を示す．

E-cube：最下位ビット優先法（E-cube 法）
SBP 1 ：SBP 法（飛び越しバンク選択）
SBP 2 ：SBP 法（逐次バンク選択）

表 5 から，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．特に，KL 1 の通信で比較的多いと予想される短いパケット長で両ルーティング法の差が小さいことが分かった．したがって，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．

5.4 ルータノードの接続プロセッサ台数

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キューブのスループットを有効利用できる．

各ルータノードにプロセッサを 4 台および 1 台接続した場合にルータから送出できる転送速度について，5.3 節と同様のシミュレーションを行った（**図 6** 参照）．図 6 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．

5.5 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．クラスタ用の 6 ポートのうち，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．ネットワークの諸元を**表 6** に示す．

表 5 ネットワークスループットとルーティング方式
Table 5 Network throughput and routing methods.
〔スループットの単位：MB/s〕

最大パケット長	E-cube	SBP 1	SBP 2
200 Byte	15.9	15.9	15.4
500 Byte	14.6	14.6	13.6
1,000 Byte	13.4	13.7	12.3
2,000 Byte	12.3	13.3	11.8
5,000 Byte	10.5	12.1	11.1

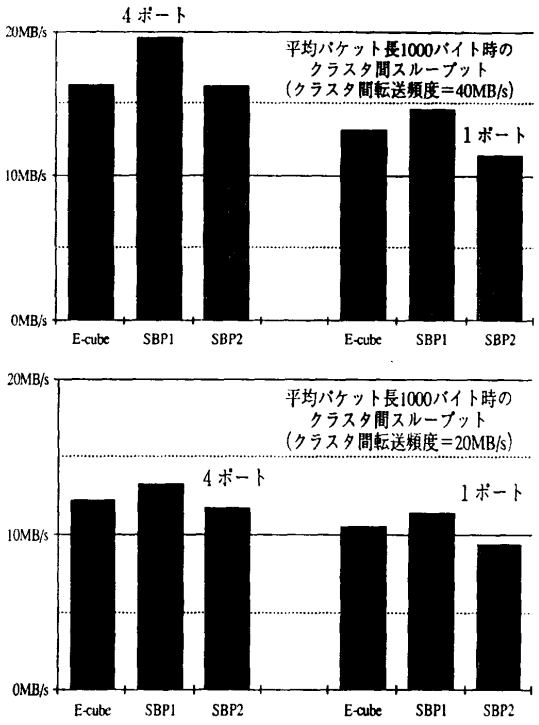

図 6 ネットワークスループットと PE ポート数/ルータ
Fig. 6 Network throughput and PE-port number/router.

表 6 クラスタ間ネットワークの諸元
Table 6 Specification of inter-cluster network.

転送速度	20 MB/s（/リンク）
クロック周期	50 ns
パケット長	8 KB（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	

6. 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シミュレーションによる評価について述べた．プロセス間の通信と同期を頻繁に行う KL 1 プログラムに適した並列アーキテクチャとして，二階層結合を採用した．局所的な高速通信のため

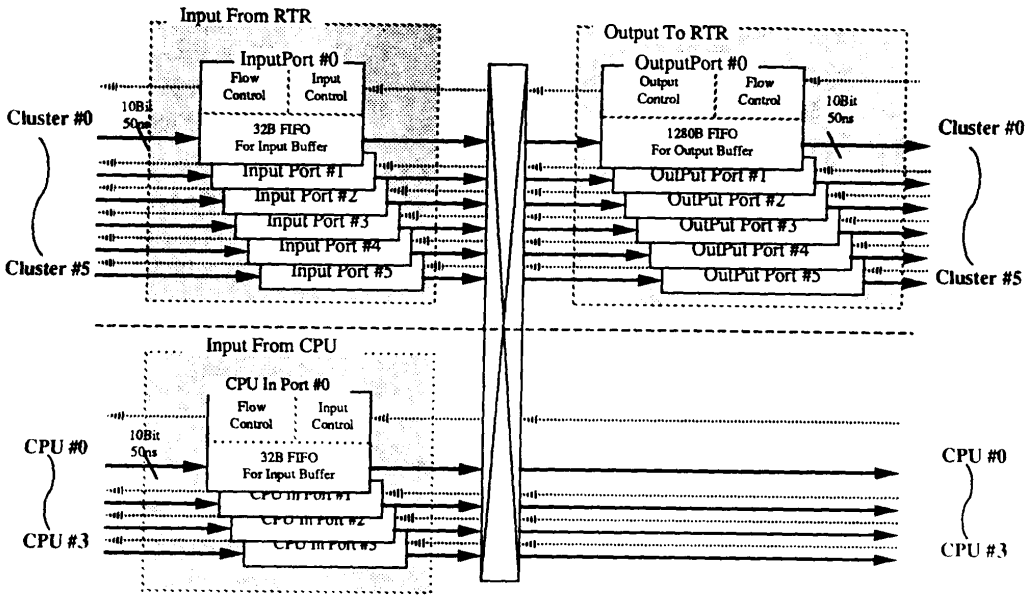

図 7 ネットワークルータの構成
Fig. 7 Configuration of a network router.

のクラスタに関して，共有バス負荷を軽減するアーキテクチャを導入してその効果を評価した．大規模化手段としてのネットワークに関して，ハイパキューブ構造のスループットの使用効率を向上する結合法の導入とデッドロックフリーな転送法の評価を行った．以上の結果，次のことが明らかになった．

（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）で十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棚橋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研究室長に感謝いたします．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 研究開発メンバー諸氏に感謝いたします．

参 考 文 献

1) 後藤厚宏，篠木 剛，久門耕一，服部 彰：並列推論マシン PIM/p の概要，第 37 回情報処理学会全国大会論文集，p. 135 (1988).
2) Goto A., Matsumoto, A., Sato, M, Nakajima, K. and Taki, K.: Overview of Parallel Inference Machine Architecture (PIM), *Int. Conf. on FGCS*, p. 208 (1988).
3) Ueda, K.: Guarded Horn Clauses: A Parallel Logic Programming Languages with the Concept of a Guard, TR 208, ICOT (1986).
4) Chikayama, T.: Multiple Reference Management in Flat GHC, *Proc. 4th ICLP*, pp. 276-293 (1987).
5) Patterson, D. and Sequin, C.: A VLSI RISC, *IEEE Comput.*, Vol. 15, No. 9, p. 8 (1982).
6) 篠木 剛，松本 明，近山 隆，後藤厚宏，服部 彰：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，第 37 回情報処理学会全国大会論文集， p. 137 (1988).
7) Shinogi, T., Kumon, K., Hattori, A., Goto. A., Kimura, Y. and Chikayama, T.: Macro-call Instruction for the Efficient KL 1 Implementation on PIM, *Int. Conf. on FGCS 1988*, p. 953 (1988).
8) Archibald, J. and Baer, J.: Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model, *ACM Trans. Comput. Syst.*, Vol. 4, No. 4, pp. 273-298 (1986).
9) Kimura, Y. and Chikayama, T.: An Abstract

KL 1 Machine and Its Instruction Set, *Proc. SLP*, pp. 468-477 (1987).
10) Merlin, P. and Schweitzer, P. : Deadlock Avoidance in Store-and-Forward Networks-I : Store-and-Forward Deadlock, *IEEE Trans. Comm.*, Vol. 28, No. 3, pp. 345-354 (1980).
11) Raubold, E. and Haenle, J. : A Method of Deadlock-Free Resource Allocation and Flow Controll in Packet Networks, *Proc. ICCC 1976*, pp. 483-487 (1976).
12) Dally, W. and Seitz, C. : Deadlock-Free Message Routing in Multiprocessor Interconnection Networks, *IEEE Trans. Comput.*, Vol. C-36, No. 5, pp. 547-553 (1987).

(平成元年5月30日受付)
(平成元年9月12日採録)

服部　彰（正会員）

昭和24年生．昭和47年大阪大学工学部電子工学科卒業．昭和49年同大学院工学研究科修士課程修了．同年(株)富士通研究所入社．階層メモリ，記号処理マシンの研究を経て，並列コンピュータの研究開発に従事．現在，人工知能研究部第3研究室長．電子情報通信学会，人工知能学会各会員．

篠木　剛（正会員）

昭和29年生．昭和52年東京工業大学情報科学科卒業．昭和54年同大学院修士課程修了．同年(株)富士通研究所入社．昭和60年9月より1年間米国Oregon大学計算機情報科学科に客員研究員として滞在．現在，富士通研究所人工知能研究部第3研究室に所属する．Lispマシン FACOM α のハードウェア，システムソフトウェアなどの開発を経て，現在，並列推論マシン PIM/p のハードウェアの研究開発に従事している．

久門　耕一（正会員）

昭和31年生．昭和54年東京大学工学部電気工学科卒業．昭和59年同大学院博士課程中退．同年(株)富士通研究所に入社．並列処理ハードウェアに関する研究に従事．非数値計算処理の並列化に興味を持つ．

後藤　厚宏（正会員）

1956年生．1979年東京大学工学部電子工学科卒業．1981年同大学院情報工学専門課程修士課程修了．1984年同博士課程修了．工学博士．同年日本電信電話公社（現NTT）研究所入社．1985年より(財)新世代コンピュータ技術開発機構へ出向．並列推論マシンの研究開発に従事．電子情報通信学会，IEEE，ACM 各会員．