Vol. 37 No. 1　　情　報　処　理　　Jan. 1996

解　説

計算物理学と超並列計算機—CP-PACS 計画—

2. 超並列計算機 CP-PACS のアーキテクチャ†

中澤 喜三郎††　中村　宏††　朴　泰祐††

1. はじめに

CP-PACS (Computational Physics by Parallel Array Computer System) は，**新プログラム制度**に基づく研究，“専用並列計算機による「場の物理」の研究”の中の1主要テーマとして，筑波大学で，平成4年度から5年間の計画で開発が進行中の超並列計算機である．この研究では，**超並列計算機の開発**と，それを用いた**計算物理学の研究**を行うことを目的としている[1]．本研究がスタートするにあたっては，先行して筑波大学で実施されていた **PAX** (Processor Array eXperiment) シリーズ，特に **QCDPAX**[2]の経験・成果が大きく寄与している．

CP-PACS プロジェクトで新たに開発を行っている超並列計算機 CP-PACS は，QCD (Quantum Chromodynamics) 計算を主要な応用問題とするが，その他の各種科学・技術計算も応用問題とする．そのため，超並列計算機開発にあたっては，計算機工学関係者のみならず，計算物理学のユーザや数値解析のアルゴリズム開発関係の研究者が共同研究を行っており，この点は大きな特徴である．この研究推進のため，全国共同利用の**「筑波大学計算物理学研究センタ」**が新たに設置され，素粒子・物性・宇宙物理と並列計算機工学の4研究部門が設けられている．現在このセンタには QCDPAX が移設され，また，CP-PACS のためのフロント計算機 (FCS) なども設置され稼働中である．

超並列計算機 CP-PACS そのものの開発に関しては，まず基本的な検討を行って概略の目標仕様イメージを定め，その後アーキテクチャ・ハードウェア・ソフトウェアの基本設計・詳細設計を行った．現在，平成7年度中の稼働を目指して開発は鋭意進行中である．それと並行して，開発設計の各段階で，計算物理学の実問題についての仮想ベンチマーク問題による性能の評価と，実用に耐え得る超並列計算機としての実現可能性との両面から検討・改善を加えてきた．ここでは，CP-PACS のアーキテクチャの概要を報告する．

2. CP-PACS のアーキテクチャ

現状での CP-PACS 全体の主要目標仕様[3]を**表-1** に示す．この章では，この表の各項目について簡単な説明を行う．主要な項目については **3.** **4.**で詳細を述べる．

(1) 全体の理論的なピーク性能

計算物理学関係者が計算機に対して要望する計算能力は，**表-2** にその例を示すように，数百 GFLOPS と厳しいものであった．

並列計算機システム全体の理論的ピーク性能は，これを構成する個々の PU (node processor) の理論的ピーク性能と，全体の PU 台数の積で表される．近年の半導体技術の進歩，および計算機設計技術の進歩により，本研究スタート時点では1台の汎用高性能マイクロプロセッサで，ピーク性能 150 MFLOPS のものが実用の域に達し始めていた．その後もマイクロプロセッサの性能が毎年同じ率で向上し続けるとすると，研究期間中に 300 MFLOPS 程度以上の性能のものが十分実現可能になると，本研究のスタート時点で予測できた．そこで，PU の性能目標として，300 MFLOPS 程度以上のものを中心に考えた．しかし，それ以上に単体の PU 性能を強化する方向は，コストパフォーマンス面での配慮からとらなかった．一方，PU の総台数については，全体と

† The Architecture of Massively Parallel Processor CP-PACS by Kisaburo NAKAZAWA, Hiroshi NAKAMURA and Taisuke BOKU (Institute of Information Sciences and Electronics, University of Tsukuba).

†† 筑波大学電子・情報工学系

しての目標ピーク性能を確保するためにPU台数もできる限り大きくしたいが，その反面，

- 並列処理の台数を増やして性能を稼ごうとしても，実効的な処理性能が上がらない場合がある，
- 完成時の装置規模，および信頼度が現実的か否か，
- 性能と，必要とされるコスト（予算）の兼ね合い

なども考慮する必要があった．そこで，CP-PACSでは，全体で1,000～2,000台程度の構成のものを中心に各種の検討を行った．その結果，相互結合ネットワークの構成方式として，後述のHXBネットワーク（**図-1**）を採用することが明確となり，加えて物理的な実装の可能性も明らかとなってきたので，当初1,024台，拡張後2,048台のPUからなるシステムとすることに，目標仕様の設定を行った．2段階としたのは予算上の都合である．

したがって，並列計算機としてのCP-PACSの目標ピーク性能（64ビットデータ）は，当初300 GFLOPS以上，拡張後600 GFLOPS以上とした．

（2）全体の主記憶容量

計算物理学上の重要な計算では，**表-2**に示すような総主記憶容量が必要となる．この要求と，使用可能な半導体記憶素子や各PUでの主記憶装置の構成方法などを考慮して，主記憶容量を64 MB/PUに設定した．

（3）並列方式

1,000台規模のプロセッサを使用するため，物理的な構成方法を考慮すると，主記憶装置は各PUに分散して実装せざるを得ない．したがって必然的に分散記憶型となる．物理的には分散して実装するにしても，プログラムから見たときには，あたかもすべてのPUが共有している記憶装置があるように見える，分散共有メモリ型の並列システムの研究も行われている．しかし，超並列機で実用に耐えるほどの十分な成果は，いまだ出ていない．そこで，CP-PACSでは，send/

表-1 CP-PACSの主要目標仕様

（1）全体の理論的なピーク性能：64ビットデータ計算時に， 当初300 GFLOPS以上， 拡張後600 GFLOPS以上
（2）全体の主記憶容量：当初64 GB， 拡張後128 GB
（3）並列方式：分散記憶型MIMD方式
（4）ノードプロセッサ(PU)：当初1024台， 拡張後2048台 PA-RISC 1.1アーキテクチャを基本に，PVP-SW (Pseudo Vector Processor based on Slide Windowed Registers) 機能付加，物理浮動小数点レジスタ数：128 (64ビット)，2命令同時発行スーパスカラ ・クロック周波数：150 MHz以上 ・キャッシュメモリ L 1：I，D各16 KB L 2：512 KB以上 ・TLBエントリ：I，D各256
（5）相互結合ネットソーク：3次元Hyper-Crossbar Network（HXB），1024台；8×17×8 2048台；8×17×16の構成 ・転送ピーク性能：300 MB/sec以上 ・転送方式：wormhole転送，高速転送プロトコル，block-stride転送 ・その他：バリア同期，放送
（6）補助記憶装置：528 GB以上（RAID-5を使用），64台のIOU（I/O unit）経由分散接続
（7）全体を小部分に分割した運用：可能とする
（8）ソフトウェア： ・言語：アセンブラ，C，Fortran，HPF ・ライブラリ：プロセス間通信用および，高速ファイル処理用など ・OS：マイクロカーネルをベース
（9）外部接続： ・FCS（front computer system）との接続：ピーク100 MB/sec以上の高速接続 ・その他：IOU経由FDDI，Ether，SCSI
(10) 稼働開始：1995年度
(11) 並列計算機信頼度：hardware MTBF＞2000時間

表-2 計算物理学 大規模計算の所要能力評価例

仮定した性能	計算例（主たる解法）	格子数	所要計算時間	所要主記憶容量	所要補助記憶容量
400 GFLOPS（実効） 20%（通信/演算） 300 MB/sec（I/O）	quenched QCD（モンテカルロ法）	64^4	68日	18 GB	216 GB
	full QCD（連立一次方程式）	24^4	58日	1 GB	3 GB
		32^4	244日	3 GB	9 GB
500 GFLOPS（実効） 100 MB/sec（I/O）	宇宙流体力学（オイラ格子法）	1000^3	120 hr	80 GB	400 GB
	電磁流体力学（Roe MHD法）	800^3	250 hr	136 GB	683 GB

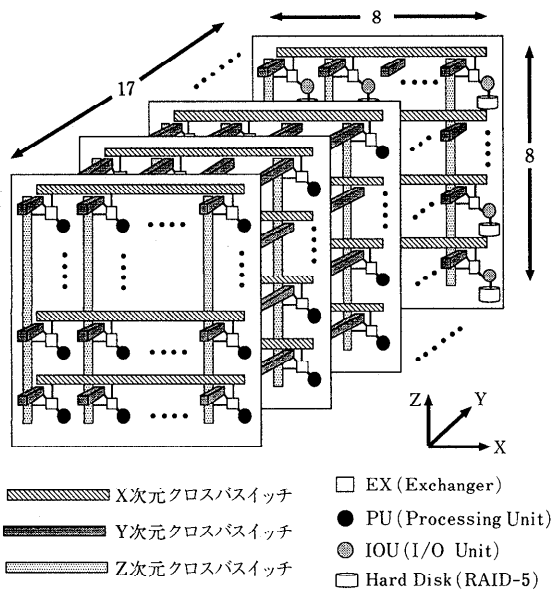


図-1 CP-PACSのアーキテクチャ
(1024 PU+64 IOU−8×17×8)

receive型のメッセージパッシング (message passing) 方式を採用した.

並列処理の方法に関しては, 特に初期においては, SPMD(single program multiple data) 方式での処理が多いものと思われ, もしそうであるならばSIMD型の並列機でもよいのではないか, という議論もある. しかし, 多種多様な応用にも応えられ, しかもOLTP(on-line transaction processing) や, データベース (data-base) 処理などの並列計算機工学上の問題にも対処することも考えて, MIMD方式を採用することとした.

(4)ノードプロセッサ

個々のノードプロセッサを構成するPUについては, 本プロジェクトの計画当初は, 汎用の高性能マイクロプロセッサをそのまま使用することを考えていた. しかし, 実際の計算物理学の応用問題での性能を検討した結果, 汎用のマイクロプロセッサそのままでは, 大規模計算の場合に十分な性能を発揮できない場合がかなりあることが判明した.

そこでCP-PACSとしては, プロセッサアーキテクチャとして, 次章で詳細に述べるような, PVP-SW(Pseudo Vector Processor based on Slide Windowed Registers, スライドウィンドウレジスタを利用した擬似ベクトルプロセッサ) 方式を新たに考案した[4)~6)].

CP-PACSでは, 汎用のスーパスカラ方式マイクロプロセッサの機能を拡張してこのPVP-SW方式を実現するものを新たに開発し, これをノードプロセッサとして採用する. この点に関しては, メーカの開発フェーズとうまくタイミングが合った点が幸いした. この機能の付加により, 各PUは実際の問題の計算時に, ピーク性能の75%以上の性能を達成できることが期待されている[7)].

(5)相互結合ネットワーク

PU間相互結合ネットワークに関しては, 隣接したPU間のデータ転送のみならず, 多様な通信パターンを持つ各種科学技術計算を効率的に処理できることを前提とし, 各種の結合網を検討した. その上で, 結合網の主要部分となるスイッチを構成するVLSIの実現可能性, および相互結合網全体の実装方法と性能との関係を検討し, 3次元のHXB[9)](Hyper-Crossbar Network) (図-1) を採用した. 1本の物理的な転送リンクの性能は, wormhole方式の転送と, 通信線上でのパイプライン式転送を行うことにより, ピーク300 MB/sec以上を目指している. ネットワークでのデータ転送方式や, 同期・放送機能については, **4.** で詳しく述べる.

上記の(3), (4), (5)を基本とする並列処理方式の採用により, CP-PACSでは, 計算物理学上の主要な計算 (たとえばQCD計算) に関し, 並列システム全体としての実効性能 (sustained speed) が, 理論ピーク性能の60%以上を達成できることが期待されている[7)].

(6)補助記憶装置

計算物理学上の重要な計算では, 表-2に示すように補助記憶容量が必要である. そこで, 補助記憶装置として, 総記憶容量528 GB以上の磁気ディスク記憶装置を, 並列計算機から直接使用できるように, 並列計算機内に分散して接続する. これらの補助記憶装置は, 長時間に及ぶ実用計算の中間結果などのデータの一時記憶用, バックアップ用, などのために用いられる. 大容量であるため, 磁気ディスク装置の台数は数百台を超えるものとなる. そこで, 信頼性に配慮して, RAID[10)](redundant array of inexpensive disk)

-5タイプのものを使用する．大量のデータの読み書きに際して，並列I/Oを行うことにより高いスループットを確保できるよう，HXBネットワークにもう1面のIOU(input/output unit) プロセッサ平面を追加し($n_x \times (n_y+1) \times n_z$の構成となる)，これを経由してRAIDを分散接続する．(図-1参照)

(7)分割運用

複数の独立な小規模な問題を処理しなければならない場合などに，あたかも複数個の独立な並列計算機が設置されているようなイメージで運用できるよう，全体を小部分に分割した運用を可能とする．

(8)ソフトウェア

ソフトウェア関係については，商用機のように完備したOS，言語処理系などのシステムソフトウェアを，CP-PACS向きに当初から揃えることは，開発力・予算から考えてほとんど不可能であった．そこで，必要最低限のシステムプログラムから開発する方針をとった．

最低必要となるプログラミング言語として，Fortran, C, およびアセンブラ言語を選び，少なくともこれらについては，PUのアーキテクチャに追加したPVP-SW機能をサポートするコンパイラ（およびWS上でのクロスコンパイラ）を実現する[11]．並列処理向きの言語としてはHPF[12](high performance Fortran) を取り上げる．

並列向きのプログラム開発環境としては，一般にも使用が始まりつつあると思われるExpressに近いもののサブセットを想定し，まずこれの整備を目指す．ワークステーション上でのデバッグのためのシミュレータの用意も行う予定である．

OSに関しては，UNIX系のマイクロカーネルをベースとしたものを備える予定である．これを整備しつつ，結合ネットワーク通信用のライブラリ群，磁気ディスク補助記憶装置用のファイルの取扱い手段，FCSとの間のファイル転送手段，ジョブやプロセスの制御手段，運用・管理ソフトウェアなどを逐次整備していくことから運用が始まるものと考えている．

ソフトウェアの詳細については，本特集の次稿[15]を参照されたい．

(9)外部接続

外部との接続として，CP-PACSとフロント計算機FCSとの間でピーク転送性能100 MB/sec以上の高速接続を行い，外部との高速大量のデータ転送手段を用意する．この接続は，ソフトウェアオーバヘッドを含めても，50 MB/sec以上の実効転送能力を達成することを目指している．また，ジョブの投入・実行・出力の制御などをFCSより一元的に行えるようにするために，FDDI, Ethernetなどの接続も行う予定である．システム全体の概略構成に関しては，前稿[21]を参照されたい．

(10)稼働開始時期

1995年度中に当初構成のシステムを稼働させ，1996年度にはソフトウェアの整備とともに，システムの拡張を行いつつ実計算も開始する予定である．

(11)並列計算機信頼度

超並列システムは，物理的規模が大きくなるので，その信頼度の確保は重要な問題となる．CP-PACSでは，最近のコンピュータ技術の発展によるスーパコンピュータや大規模汎用システムの稼働実績を考慮して，ハードウェアのMTBFが2000時間以上であることを目標とする．

3. 擬似ベクトルプロセッサPVP-SW

3.1 方　　針

超並列計算機のノードプロセッサとして有力な候補の1つは，近年性能向上が著しい汎用の高速RISCマイクロプロセッサである．しかし，その高速性は「キャッシュが有効に働く」場合においてのみ達成される．大規模な科学技術計算においては，データ領域が非常に大きく，またデータの時間的局所性が少ないという性質があるため，データキャッシュのミス率が大きい．このため，通常の汎用RISCプロセッサの実効的な性能は，主記憶アクセスレーテンシによるペナルティのために大きく低下する[13]．

一方，科学技術計算において高い処理性能を有するベクトルプロセッサは，以下の3つの特徴を持つ．(1)主記憶アクセスがパイプライン化され，主記憶のスループットが高い，(2)多数のベクトルレジスタを持つためベクトルロード/ストア処理のベクトル長を長くできる，(3)チェイニング機構によりロード/ストア処理と演算処理と

を並行に行える．これらの特徴により，ベクトルレジスタへのプリロード機能を実現でき，主記憶アクセスレーテンシが性能に与える影響を隠蔽することができる．

そこで，CP-PACSのノードプロセッサではこれと同等な機能/特徴をスーパスカラ方式のプロセッサに機能拡張を施して実現する．この方式を我々は**擬似ベクトル処理**（Pseudo Vector Processing)[4]と呼ぶ．擬似ベクトル処理を実現するためには，上記特徴に対応する以下の3点を実現する必要がある．

- ●主記憶アクセスのパイプライン化
- ●多数の浮動小数点レジスタ
- ●レジスタへのプリロード機能の強化

我々の方針は，ソフトウェアの開発コストを抑えるべく，既存のRISCアーキテクチャとの上位互換性を保つことである．しかし，第2の点を実現するためには，命令フォーマット中のレジスタ指定フィールドのビット数を増やす必要があり，通常は命令セットアーキテクチャの互換性は失われる．そこで，我々は「擬似ベクトルプロセッサ**PVP-SW** (**P**seudo **V**ector **P**rocessor based on **S**lide-**W**indowed Registers)」を提案した[5),6)]．

3.2 擬似ベクトルプロセッサPVP-SW

PVP-SWのアーキテクチャは既存のスカラプロセッサに対する拡張として定義される．拡張点は，浮動小数点レジスタのスライドウィンドウ化と数種類の命令追加である．

(1)スライドウィンドウ構成

図-2に，浮動小数点レジスタのスライドウィンドウ構成を示す．物理的なレジスタ空間は複数の論理的なウィンドウに分割される．1つの論理的なウィンドウ内のレジスタ数は，拡張前のアーキテクチャで定義される数（図-2では32とした）と等しくなければならないが，総物理レジスタ数 m はハードウェアが許す限り任意の個数実装することが可能である．

ウィンドウの位置は**window-offset**で与えられる．各ウィンドウはglobal部とlocal部にわかれ，global部のレジスタはすべてのウィンドウに共通である．図-2では，global部のレジスタ数を8とした．

ある時点では，ただ1つのウィンドウのみがアクティブであり，このアクティブウィンドウを指

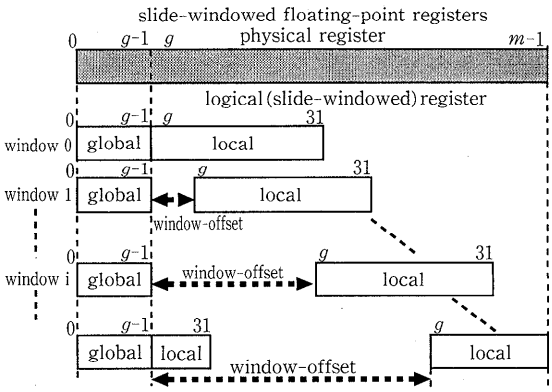


図-2　PVP-SWにおける浮動小数点レジスタの構成

定するポインタ**FWSTP**(**F**loating-point **W**indow **ST**art **P**ointer）を導入する．拡張前のアーキテクチャにおける命令は，すべてこのアクティブウィンドウ内のレジスタのみを用いる．一方，ソフトウェアでFWSTPの値を変更することにより，このアクティブウィンドウは物理的ウィンドウ空間内を自由に移動（スライド）可能である．このため，上位互換性を維持しながら多数の物理的なレジスタを利用することが可能となる．

(2)追加命令

以下の命令が追加される．

FWSTPset　アクティブウィンドウの位置を指定するFWSTPの値を変更し，アクティブウィンドウを切り替える．

FRPreload　データをメモリから任意に指定されるウィンドウ内のレジスタにロードする．

FRPoststore　データを任意に指定されるウィンドウ内のレジスタからメモリにストアする．

FRPreloadおよびFRPoststore命令は主記憶とレジスタとのデータ転送を行う命令であるが，以下の3つの特徴を持つ．

(1)「置いてきぼり処理」により，データ転送の完了を待たずに後続命令は実行される

(2)アクティブウィンドウに関係なく，全物理レジスタをデータ転送の対象として指定できる

(3)キャッシュミス時には主記憶とレジスタの間で直接データを転送し，キャッシュ内データを変更しない

1番目の特徴により，将来必要となるデータに対するプリロード命令を十分前もって発行することで，他の命令実行を妨げることなく該当データ

に対する主記憶アクセスレーテンシを隠蔽できる．より長いレーテンシを隠蔽するためには，それだけプリロード命令を早く発行する必要があり，必要なレジスタ数が増加する．この問題は物理的に多数のレジスタを用意すること，および2番目の特徴で解決する．3番目の特徴は，キャッシュのスループットが十分でない場合に性能が低下するのを防ぐため，およびキャッシュコヒーレンシ問題に対処するためである（この点に関しては**4.2**に詳述する）．これらの特徴により，PVP-SW方式が主記憶アクセスレーテンシを効果的に隠蔽しきわめて高い性能を示すことを，各種ベンチマークの評価によって確認している．

3.3 PVP-SWにおける擬似ベクトル処理例

（1）オブジェクトコードとレジスタ割当て

DAXPYループ‘$Y(i)=a\times X(i)+Y(i)$’を例として，PVP-SWにおける擬似ベクトル処理を説明する．**図-3**にオブジェクトコードを示す．このコードは，modulo schedulingを用いたソフトウェアパイプライン手法により生成している．アルゴリズムの詳細については本特集の次稿[15]を参照されたい．

```
Loop : FRPreload     Y(i+1)->r30<+2>
       ADD           r29+r30->r29      %aX(i)+Y(i)
       FRPreload     X(i+2)->r31<+2>
       MULT          r7 * r31->r31     % a * X(i+1)
       FRPoststore   r29<-0>   >new Y(i)
       FWSTPset+2    (FWSTP<-FWSTP+2)
       % branch is omitted
```

図-3 PVP-SWにおけるDAXPYループのコード例

図-3では，説明を簡単にするため，分岐命令は省略してある．図-3中の‘FRPreload Y(i+1)->r30<+2>’は，現在アクティブであるウィンドウの2つ先のウィンドウ（そのウィンドウのwindow-offsetは，アクティブウィンドウのwindow offsetより2大きい）の論理レジスタr30にY(i+1)のデータをプリロードすることを表す．また，最後にある‘FWSTPset +2’はFWSTPの値を2増やすことを表す．

図-3のオブジェクトコードを実行するときのレジスタ割当てを**図-4**に示す．図-4ではループ処理が時間の経過と共に展開されて示されている．6 cycleごとの点線は，アクティブウィンドウが切り替ることを表す．影付きの部分はY(i+2)の計算に関係する部分である．

ソフトウェアパイプライン手法を用いているため，図-4より分かるように，たとえばY(i+2)の計算は複数のループにまたがって処理される．一般には，ソフトウェアパイプラインを用いると，ループ間でWARハザードが発生する場合がある．レジスタリネーミング機構がハードウェア的に実現されていない場合には，このハザードの解消のために，modulo variable expansion[17]

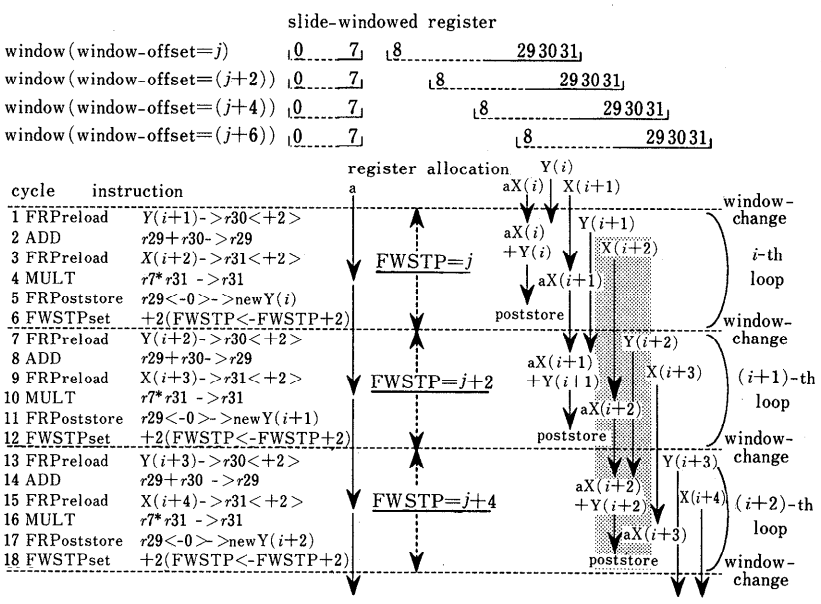


図-4 PVP-SWにおけるレジスタ割当て

などの手法によりコンパイラがソフトウェア的に対処する必要がある．

しかしPVP-SWでは，ループごとにウィンドウが切り替えられるため，各ループは異なるウィンドウ上で処理されることになり，同一の論理番号を持つレジスタも，ループごとに物理的には異なるレジスタに対応することになる．これはすなわち，PVP-SWではレジスタリネーミングと等価なことが自然に実現されていることになり，上記のWARハザードに対する特別な対処を必要としない．

(2)主記憶アクセスレーテンシの隠蔽

あるデータに対するロード要求から，そのデータが演算命令により最初に使用されるまでの時間間隔を，そのデータに対する *permitted latency* と呼ぶ．たとえば図-4ではX(i+2)に対する *permitted latency* は7 cycleである．すべてのデータに対する *permitted latency* が主記憶アクセスレーテンシ以上であれば，主記憶アクセスレーテンシによる性能低下を隠蔽することができる．たとえば，図-4において，*permitted latency* の最小値は7 cycleであるので，7 cycle以下の主記憶アクセスレーテンシを隠蔽できることになる．より長い主記憶アクセスレーテンシを隠蔽するためには，*permitted latency* を長くする必要がある．そのためにはより遠くのウィンドウに，より早くデータをプリロードする命令を実行させればよい．具体的には，図-3における'FRPreload Y(i+1)->r30<+2>'および'FRPreload X(i+2)->r31<+2>'の下線部を，たとえば，'FRPreload Y(i+2)->r30<+4>'および'FRPreload X(i+3)->r31<+4>'のように変更すればよい．この変更では，プリロード命令がさらに1ループ先行して発行されることになるため，結果的に *permitted latency* は，

7 (変更前の *permitted latency*)+

6 (1ループの実行に要する時間) =13 cycle

になる．

どれだけ遠くのウィンドウにプリロードできるかは，総物理レジスタ数によって制限されるため，隠蔽可能な主記憶アクセスレーテンシの長さも総物理レジスタ数により制限される．これまでの評価では，2命令同時発行のスーパスカラ方式を仮定すると，物理的に128個のレジスタがあれば，100マシンサイクル程度のアクセスレーテンシを隠蔽できることが分かっている[16]．

3.4 CP-PACSで採用予定のプロセッサ

CP-PACSでは，Hewlett Packard社のPA-RISC 1.1アーキテクチャを基本に上記のPVP-SW機構を追加したものを，ノードプロセッサとして採用する．PA-RISC 1.1は32本の倍精度浮動小数点レジスタを持っており，これを拡張して物理レジスタ群とウィンドウ制御回路（主に論理レジスタ番号と物理レジスタ番号の変換をする）を追加する．実装する物理レジスタの数は128本を予定している．また，グローバル・レジスタの本数はソフトウェア的にある程度の範囲で可変に制御できるように考えている．

プロセッサチップはメーカと協力して開発中であり，現時点では以下のような実装になることが予定されている．

テクノロジ 0.3 μm, 4-level metal CMOS technology,

電源電圧と消費電力 VDD: 2.5 V, Power: 13 W at 150 MHz,

トランジスタ数とチップ面積 4.5 M transistors, 15.7 mm×15.7 mm die size,

パッケージング : 1672-pin CCB (controllable collapsible bonding), 信号520 pin.

4. ハイパクロスバ・ネットワーク

4.1 HXBとその特徴

CP-PACSにおけるPUおよびIOU間の結合はHyper-Crossbar Network（以下HXBと省略）を用いる．HXBのトポロジは，n 次元の超立方体の概念を拡張した形としてとらえることができる．ここで次元数 n は任意に選択可能であるが，CP-PACS実装時の技術を想定すると，3次元の構成が妥当であると考えられる．

一般的な3次元HXBの特徴について，$X\times Y\times Z$ 構成の場合を想定して説明する（図-1は8×17×8の構成）．まず，全PUは $X\times Y\times Z$ の3次元正方格子状に配置される．そして，1つの次元方向（たとえば x 次元方向）に並んだPU同士をクロスバ・スイッチ（以下，XBと省略）で完全結合する．たとえば x 次元方向の場合，X 入力 X 出力のXBが必要となる．このようにしてすべての次元について，その次元方向に並ん

だPU間をXBで結合する．この場合，次元数 n は3なので，各PUは3つの次元方向のXBに対するリンクをそれぞれ持つことになる．

メッセージ転送において，経路決定法を固定とするルーティング・アルゴリズムは単純である．$PU(x_s, y_s, z_s)$ から $PU(x_d, y_d, z_d)$ までメッセージを転送する場合，まず $PU(x_s, y_s, z_s)$ から $PU(x_d, y_s, z_s)$ までメッセージを送り，続いてそれを $PU(x_d, y_d, z_s)$ へ，そして最後に $PU(x_d, y_d, z_d)$ へ，というように3ステップの転送を繰り返して送ればよい．CP-PACSではこの際に転送遅延を極力小さくするためにwormhole方式のメッセージ転送を行う．このため，1つのPUと各次元方向のXBを結合するために，EX(Exchanger）と呼ばれるルータ・スイッチを設ける．EX自身も小規模のXBである（たとえば3次元の場合，EXは4×4のXBである）．

wormhole方式のHXBには，

- ネットワークの半径が小さい
- 同一サイズあるいはそれ以下のサイズのトーラス・ネットワークをエミュレートできる
- Binary-n-Cubeネットワークのエミュレートが容易である
- ハードウェアによるブロードキャスト転送が容易に実現できる
- ランダム転送におけるスループットが非常に高い

などの特徴がある[9]．

wormhole方式のHXBでは，一般的に経路決定を動的に行うとデッドロックが生じるが，各通信チャネルに2〜n 本のバーチャル・チャネルを用意すれば，経路決定を動的に行う適応ルーティングも可能である[18),19)]．しかし，CP-PACSにおいては，科学技術計算における定型的なPU間交信がアプリケーションの多くを占めることと，通信チャネルをバーチャル化するためのコストを考慮して，固定ルーティング方式を用いている．

以下に，CP-PACSにおけるPU間結合ネットワークの特徴をいくつかあげる．

4.2 リモートDMA転送

CP-PACSでは，メッセージ転送における高速立ち上げと，高スループットを実現するために，ユーザ・アプリケーションから直接ネットワークへのメッセージ転送起動が行えるような機構を提供する．これをリモートDMA転送と呼ぶ（**図-5**）．

一般的なメッセージ送受信プリミティブでは，まず送信側プロセスのユーザメモリ空間上のデータがOS内の通信バッファにコピーされ，それがNIA（Network Interface Adapter）を通じて受信側PUの通信バッファに転送され，最終的に受信プロセスが受信命令を発行することにより，データが通信バッファからユーザメモリ空間に再度コピーされて受信処理が完了する．これに対し，リモートDMA転送では，送受信プロセスのユーザメモリ空間内で，データが直接メッセージとして転送される．さらに，NIAに対する送信起動操作はシステムコールを用いず，ある制約の下で，ユーザアプリケーションから直接起動できるようにする．これらの機能により，メモリ上での冗長なデータ・コピーが排除され，実効転送スループットが向上し，さらにメッセージ転送時の立ち上げオーバヘッドも大幅に短縮される．

リモートDMA転送では，受信側プロセスにおけるデータ格納番地が送信側プロセスによって

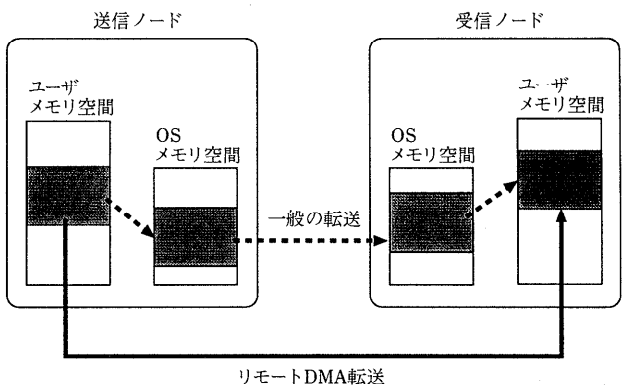


図-5 リモートDMA転送の概念図

指定されるため，送信側プロセスは受信側プロセスのアドレス空間上での変数の配置を知っている必要があるが，CP-PACSでは一般にSPMD方式のプログラミング・モデルを用いるため，問題は生じない．また，リモートDMA転送はあらかじめプログラムの初期段階での一括したOSに対するサービス要求により，送り先PUやプロセスに対するデータ転送許可を得てから実行されるため，マルチユーザ/マルチプロセス環境における安全性は保証される．

CP-PACSが対象とする科学技術計算では，あらかじめ定められたPU間での，大量のデータ転送が頻繁に行われる．リモートDMA転送における個々のデータの送受信時のデータ・コピーの排除はこの点で非常に有利であり，分散メモリ方式アーキテクチャにおいて問題となるメッセージ転送に要するコストを大幅に短縮できる．CP-PACSではこのほかに，通常のsend/receive型のメッセージ転送プリミティブも用意される．

メッセージ転送においては，一般的に，外部から主記憶に転送されるデータと，キャッシュ上のデータとのコヒーレンスの問題が生じる．CP-PACSにおけるリモートDMA転送では，メッセージ転送時にキャッシュとの整合性をとりつつデータを主記憶に格納するモードと，キャッシュを無視してデータを格納するモードの2種類が用意され，プログラム上で切り替えることが可能である．前者の方式は安全であるが，整合性のチェックのためにデータの転送スループットが低下する可能性がある．しかし，CP-PACSにおける科学技術計算は，基本的にPVP-SW機能を用いて処理されるため，対象となるデータをキャッシュ上に持たず，擬似ベクトル処理によって主記憶とプロセッサの間で直接処理することが可能である．この前提の場合，ほかのPUから転送されたデータはキャッシュとの整合性を考えず，主記憶上に置いておくことができる．同様に，PVP-SWによって処理されたデータは主記憶上に直接置かれ，リモートDMA転送によってほかのPUに送られていく．このように，リモートDMA転送とPVP-SWは，キャッシュに依存することなしに一連の処理を行うことが可能となっており，両者の組合せによって，演算・転送処理において最大のスループットを得ることが可能になっている[20]．

4.3 ブロードキャスト転送

並列処理においてしばしば現れる特殊なメッセージ交換の例として，ブロードキャスト通信がある．CP-PACSではこれをハードウェア的にサポートする．HXBにおいては，あるXBの入力に届いたメッセージを，そのXBの全出力に同時に送出することにより，単一次元内でのブロードキャストが容易に実現できる．これを全次元に対して行えば，全PUに対するブロードキャストが実現できる．ただし，HXBでは複数PUから別々のブロードキャスト要求が同時に出た場合，デッドロックを生じるため，たとえばあるPUだけがブロードキャスト・メッセージを流すことができるという制限を設ける必要がある．現在は基本的にこの手法を元に設計を行っている．

CP-PACSでは，PU空間全体をハードウェア的に分割（パーティショニング）することが可能となっており，その場合，ブロードキャスト転送もその分割空間で閉じるようになっている．これにより，ブロードキャスト転送を必要とする複数のアプリケーションを，異なるパーティションでそれぞれ独立に同時実行することが可能となっている．

4.4 ブロック・ストライド転送

一般に，多次元配列データの一部をPU間で交換するような場合，単純な連続領域のデータ転送ではうまく行かない場合が多い．このような場合にはブロック・ストライド転送が用いられる．一般のメッセージ転送プリミティブでは，通信ライブラリなどのレベルで，ユーザのデータをOSに受け渡す際にこのようなストライド・データのパッキング/アンパッキングを行うことが可能であるが，リモートDMA転送を用いた場合，これは不可能となる．

このため，CP-PACSではこれをNIAで自動処理するようになっている．NIAではストライド・アクセスをしたデータをまとめてメッセージとして転送するため，ネットワーク転送に要する時間は実際に送信するデータ量の分だけで済む．そして，受信側のNIAによって，再びユーザメモリ空間上にストライド・データとして展開される．

ブロック・ストライド転送がハードウェアによって実現されることにより，CP-PACSでは自由

なデータのマッピングやレイアウトが可能となり，HXBの柔軟な構造と相まって，問題の並列処理に最適なアルゴリズムを選択することを容易にしている[20]．

4.5 バリア同期機構

一般的に，メッセージ・パッシング系のPU間交信では，メッセージの送受信処理そのものがpoint-to-pointの同期操作を同時に実現しているため，その他の同期処理が不要な場合が多い．しかし，現実的には並列プログラムのデバッグ時や，システムの初期化などにおいて，一斉同期が必要な場合も多い．そこで，CP-PACSではこのような場合に簡単に全PUでバリア同期をとることができるような機構を用意している．CP-PACSにおける同期ネットワークは，基本的にデータ転送ネットワークを部分的に用いることにより実装される．これは各XBにおいて同期用メッセージの待ち合わせと合流を制御することによって実現される．

先述のブロードキャスト転送と同様，バリア同期機構もパーティション内で閉じるようになっており，バリア同期を用いるアプリケーションを，パーティションごとに各々独立に同時実行することが可能となっている．

5. おわりに

CP-PACSは，計算物理学の問題を処理するために，筑波大学で開発中の超並列計算機であり，一般の科学技術計算も対象としている．本報告では，CP-PACSが目指しているアーキテクチャの概要を述べた．特徴的な点として，1）擬似ベクトルプロセッサ（PVP-SW）方式を使用しノードプロセッサの実効的な高速化を計っていること，2）ハイパクロスバ結合網（IIXB）により高効率のデータ転送を行っていること，3）大容量の補助記憶（磁気ディスク）を分散接続し高速並列I/Oを行うこと，などがあげられる．

CP-PACSシステムは，完成時，物理的には消費電力：約390 KVA，設置床面積：9 m×3.5 m以下を見込んでいる．開発は，平成7年度中の稼働開始を目指して現在進行中である．一方，実計算での性能評価仮想ベンチマーク，各種のシミュレータによる評価，並列計算用アルゴリズムの検討，応用プログラムの検討も行っている．なお，具体的な装置などの製作については，開発着手時に，製作委託を受ける可能性のあるメーカ16社（内海外6社）に，全体計画概要を示して提案を求め，最終的に入札により(株)日立製作所を選定している．

本研究は文部省科学研究費・創成的基礎研究費（07 NP 0401）によって行われているものである．本研究を推進するにあたっては，文部省・関係大学・機関・筑波大学の広範の関係者のサポートをいただいている．他の同様プロジェクトの方々からも有益なご示唆をいただいている．ここに感謝の意を表する．また日頃協力をいただいている(株)日立製作所の関係者にも謝意を表する．

参考文献

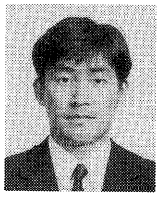
1) 筑波大学計算物理学研究センター：専用並列計算機による「場の物理」の研究（新プログラムによる研究，研究進捗状況報告書）(Aug. 1994).
2) Shirakawa, T., Hoshino, T., Oyanagi, Y., Iwasaki, Y., Yoshie, T., Kanaya, K., Ichii, S. and Kawai, Y.: QCDPAX-An MIMD Array of Vector Processors for the Numerical Simulation of Quantum Chromodynamics, Proc. Supercomputing'89, pp. 495-504 (Nov. 1989).
3) 中澤喜三郎，朴 泰祐，中村 宏，中田育男，山下義行，岩崎洋一：CP-PACSのアーキテクチャの概要，情報処理学会研究報告，94-ARC-108, pp. 57-64 (1994).
4) Nakazawa, K., Nakamura, H., Imori, H. and Kawabe, S.: Pseudo Vector Processor based on Register-Windowed Superscalar Pipeline, Proc. Supercomputing'92 (IEEE/ACM), pp. 642-651 (Nov. 1992).
5) Nakamura, H., Imori, H., Nakazawa, K., Boku, T., Nakata, I., Yamashita, Y., Wada, H. and Inagami, Y.: A Scalar Architecture for Pseudo Vector Processing based on Slide-Windowed Registers, Proc. Intl. Conf. on Supercomputing'93 (ACM), pp. 298-307 (1993).
6) 位守弘充，中村 宏，朴 泰祐，中澤喜三郎：スライドウィンドウ方式による擬似ベクトルプロセッサ，情報処理学会論文誌，Vol. 34, No. 12, pp. 2612-2623 (1993).
7) 青木慎也，金谷和至，吉江友照：超並列計算機CP-PACSの計算物理学分野における実効性能の予測，情報処理，Vol, 37, No. 1, pp. 38-42 (Jan. 1996).
8) Rau, B. R., Lee, M., Tirumalai, P. P. and Schlanker, M. S.: Register Allocation for Software Pipelined Loops, Proc. of the ACM SIGPLAN 92 Conf. on Programming Language Design and Implementation, pp. 283-299 (June 1992).

9) 朴 泰祐，斎藤哲也，板倉憲一，中澤喜三郎，中村 宏：ハイパクロスバ・ネットワークの性能評価，電子情報通信学会技術研究報告，CPSY-93-40, pp. 41-48 (1993).
10) Katz, R. H., Patterson, D. A. and Gibson, G. A.: Disk System Architecture for High Performance Computing, Proc. IEEE, 78: 2 (Feb. 1990).
11) 山下義行，中田育男：ループ中に条件分岐を含む場合の最適なソフトウェア・パイプライニング，並列処理シンポジウム JSPP' 94 論文集，pp. 17-24 (1994).
12) HPF: High Performance Fortran Language Specification, ver, 1.0 (1993. 5. 3. 9 PART I), Fortran Forum, 12: 4 (Special Issue) (1993).
13) Simmons, M. L. and Wasserman, H. J.: Performance Evaluation of the IBM RISC System/6000: Comparison of an Optimized Scalar Processor with Two Vector Processors, Proc. of Supercomputing'90, pp. 132-141 (1990).
14) Chen, T. and Baer, J. L.: A Performance Study of Software and Hardware Data Prefetching Scheme, Proc. of 21th ISCA, pp. 223-232 (1994).
15) 中田育男，山下義行，小柳義夫：超並列計算機 CP-PACS のソフトウェア，情報処理，Vol. 37, No. 1, pp. 29-37 (Jan. 1996).
16) Nakamura, H. et al.: Evaluation of Pseudo Vector Processor based on Slide-Windowed Registers, Proc. of HICSS-27, pp. 368-377 (Jan. 1994).
17) Lam, M.: Software Pipelining: An Effective Scheduling Technique for VLIW Machines, Proc. of ACM SIGPLAN'88 Conf. on Programming Language Design and Implementation, pp. 318-327 (1988).
18) 朴 泰祐他：ハイパクロスバ・ネットワークにおける転送性能向上のための手法とその評価，並列処理シンポジウム JSPP' 94 論文集，pp. 129-136 (May 1994).
19) 曽根 猛 他：ハイパクロスバ・ネットワークにおける virtual channel の動的選択による適応ルーティング，並列処理シンポジウム JSPP' 95 論文集，pp. 249-256 (May 1995).
20) Itakura, K. et al.: Preliminary Evaluation of NAS Parallel Benchmarks on CP-PACS, Proc. PERMEAN'95, pp. 68-77 (Aug. 1995).
21) 岩崎洋一 他：計算物理学と CP-PACS 計画，情報処理，Vol. 37, No. 1, pp. 11-17 (Jan. 1996).

(平成 7 年 9 月 22 日受付)

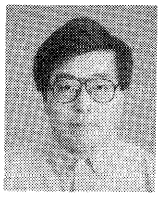
中澤喜三郎（正会員）

昭和 30 年東京大学工学部応用物理卒業．昭和 35 年同大学院数物系博士課程応用物理修了．同年日立製作所入社．TAC, HITAC 5020, E/F, 8800/8700, M-200 H/280 H, 680 H, S-810 など，超大型コンピュータ・スーパコンピュータの開発に従事．平成元年より筑波大学電子・情報工学系教授，計算物理学研究センタ向きの超並列処理システム CP-PACS の研究開発に従事．工学博士．平成 5 年度本会論文賞受賞．電子情報通信学会，IEEE，ACM 各会員．



中村 宏（正会員）

昭和 38 年生．昭和 60 年東京大学工学部電子工学科卒業．平成 2 年同大学院工学系研究科電気工学専攻博士課程修了．工学博士．同年筑波大学電子・情報工学系助手．平成 3 年同講師，平成 7 年同助教授，現在に至る．計算機アーキテクチャ，並列処理，計算機の上位レベル設計支援に関する研究に従事．本会平成 5 年度論文賞，本会平成 6 年度山下記念研究賞各受賞．電子情報通信学会，IEEE，ACM 各会員．



朴 泰祐（正会員）

昭和 59 年慶應義塾大学工学部電気工学科卒業．平成 2 年同大学院理工学研究科電気工学専攻後期博士課程修了．工学博士．昭和 63 年慶應義塾大学理工学部物理学科助手．平成 4 年筑波大学電子・情報工学系講師，平成 7 年同助教授，現在に至る．超並列計算機アーキテクチャおよび並列処理言語・アルゴリズムの研究に従事．電子情報通信学会会員．