可変構造型並列計算機の PE 間メッセージ通信機構†

森 眞一郎†† 蒲池恒彦†† 濱口一正††*
村上和彰†† 福田晃††
末吉敏則††** 富田眞治††

『可変構造型並列計算機 (Reconfigurable Parallel Processor)』と呼ぶ汎用／多目的の高速中規模マルチプロセッサ・システムを開発している．本システムは，128台のプロセッシング・エレメント (PE) を128×128のクロスバー網で相互結合した MIMD 型のマルチマイクロプロセッサ・システムである．各 PE には，マイクロプロセッサおよび浮動小数点演算プロセッサとして SPARC チップセットを搭載し，システム全体の最大性能として 1.28 GIPS および 205 MFLOPS を予定している．また，相互結合網およびメモリにダイナミック・アーキテクチャを適用し，プロセッサ-プロセッサ結合およびプロセッサ-メモリ結合に対して任意の結合形態が実現可能である．これにより，本システムは，解くべき問題の構造に合わせて結合形態を柔軟に適応させる適応型並列計算機として機能する．本論文では，この相互結合網およびメモリの可変構造化を可能とする PE 間メッセージ通信機構について述べる．PE 間メッセージ通信機構は，クロスバー網および各 PE のメッセージ通信ユニット (MCU) から構成され，プロセス間メッセージ交換機能およびリモートメモリ・アクセス機能を提供する．システム全体で最大 2.13G バイト／秒の PE 間通信バンド幅を提供すると同時に，多様な PE 間接続形態を実現可能とする．

1. はじめに

VLSI 技術の著しい発達に伴い，その特長を最大限に活用し得る新しい計算機アーキテクチャとして，マルチプロセッサ構成による並列計算機システムが注目を集めている[1]．マルチプロセッサのシステム構成にあたっては，構成要素であるプロセッサ周辺の設計もさることながら，これらをどのように相互結合するかがシステム成功の鍵を握っている．これまでに，バス，木状網，格子網，超立方体網，オメガ網などの種種の相互結合網を有した特徴のあるシステムが研究開発されている．しかしながら，これらの相互結合網は構造が静的に定まっており，一般に特定の処理形態を強く反映しているため，システムとしての応用分野が限定されがちである．また，並列アルゴリズムを作成する際にも，物理的な結合形態を常に意識しなければならないという問題点がある．

将来における処理の高並列化を促進するには，応用分野の裾野を広げ，かつ，種々の並列アルゴリズムを容易に実現できる汎用／多目的な高並列処理計算機の実現が強く望まれる．そこで，筆者らは，各種の応用問題ならびに並列アルゴリズムに対して柔軟に計算機構成を適応させる可変構造型並列計算機の開発を進めている[2]．本システムではこの目的のために，その相互結合網およびメモリにダイナミック・アーキテクチャ[7]を採用し，次のような可変構造化を試みている[4]．

① 可変構造型相互結合網アーキテクチャ：プロセッサ-プロセッサ結合およびプロセッサ-メモリ結合として任意の結合形態を可能とするよう，結合形態を限定しない極めて能力の高い相互結合網を実現する．

② 可変構造型メモリアーキテクチャ：プロセスレベルでは任意の仮想メモリイメージ，またプロセッサレベルでは任意の実メモリイメージを与えるよう，柔軟なアドレッシング機構を実現する．

本論文では，まず可変構造型並列計算機のシステム構成を述べたあと，可変構造型相互結合網および可変構造メモリアーキテクチャを実現する PE 間メッセージ通信機構について，その構成，機能および予測性能を述べる．

2. システム構成

可変構造型並列計算機のシステム構成を図1に示す．本システムは MIMD 型のマルチマイクロプロセッサ・システムであり，N台のプロセッシング・エレメント (PE) を S 台の $N\times N$ クロスバー網で相互結

† Inter-PE Communication Subsystem of the Kyushu University Reconfigurable Parallel Processor by SHIN-ICHIRO MORI, TSUNEHIKO KAMACHI, KAZUMASA HAMAGUCHI, KAZUAKI MURAKAMI, AKIRA FUKUDA, TOSHINORI SUEYOSHI and SHINJI TOMITA (Department of Information Systems, Interdisciplinary Graduate School of Engineering Sciences, Kyushu University).

†† 九州大学大学院総合理工学研究科情報システム学専攻
* 現在 キヤノン(株)
** 現在 九州工業大学情報工学部

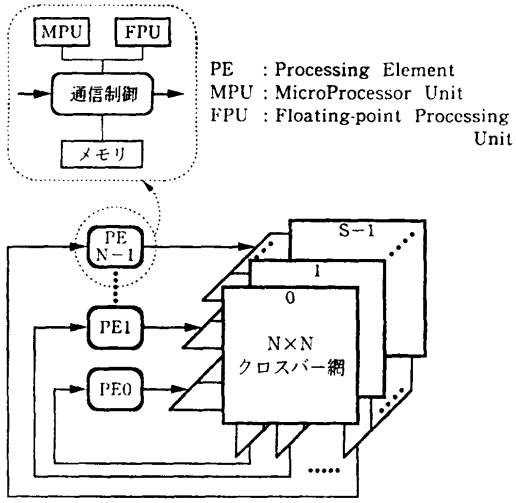


図 1 可変構造型並列計算機のシステム構成
Fig. 1 Configuration of reconfigurable parallel processor.

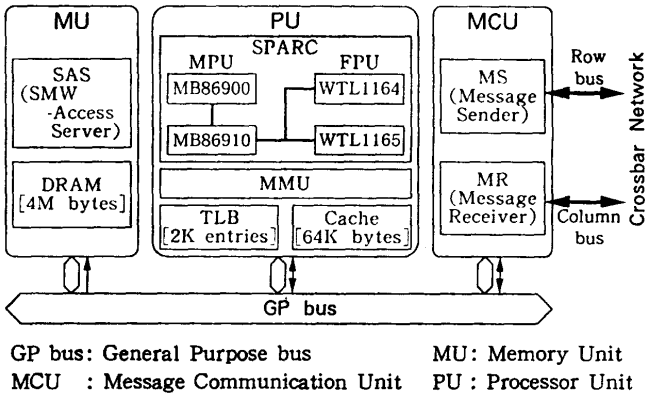


図 2 PE のブロック図
Fig. 2 Block diagram of PE.

合したものである．現在，$N=128$, $S=1$ である．

2.1 プロセッシング・エレメント (PE)

各 PE は図2に示すように，プロセッサユニット (PU)，メッセージ通信ユニット (MCU) およびメモリユニット (MU) の3つの主要ユニットから構成され，各ユニットはバス接続される．

(1) プロセッサユニット (PU)

PU は，SPARC チップセット (MB 86900: RISC マイクロプロセッサ，MB 86910: 浮動小数点演算コントローラ，WTL 1164/1165 チップセット: 浮動小数点演算器)，メモリ管理ユニット (MMU)，2048 エントリの TLB (Translation Lookaside Buffer) および 64 KB のキャッシュメモリを備える[3),5)]．動作周波数は 16.7 MHz で，整数演算性能 10 VAX MIPS，浮動小数点演算性能 1.6 MFLOPS (単精度LINPAC) および 1.1 MFLOPS (倍精度 LINPAC) の処理能力を有する．

(2) メッセージ通信ユニット (MCU)

MCU は，他 PE とのメッセージ通信を行うためのユニットであり，PU とクロスバー網とのインタフェースとして PU と並列に通信処理を行う．MCU はメッセージの送信処理を行うメッセージセンダ (MS) と受信処理を行うメッセージレシーバ (MR) とから成り，互いに独立に通信処理を行う．MCU については，4章で詳述する．

(3) メモリユニット (MU)

MU は，DRAM 構成，4 M バイト容量のローカルメモリである．さらにローカルメモリを他 PE と共有可能とするための機構 (SAS: Shared-memory-window Access Server) を MU 内に装備し，後述の共有メモリウィンドウ (SMW: Shared Memory Window) アクセスの際のアドレス変換，および，キャッシュコヒーレンスの保証を行う．

2.2 相互結合網

相互結合網を可変構造化するという要件に加え，性能および制御の容易さを考慮した結果，本システムでは相互結合網としてクロスバー網を採用した[2)]．クロスバー網には，

① PE 間の結合形態に一切制限がない，
② 非閉塞網である，
③ ルーティングが直接的で単純である，
④ PE 間距離がスイッチ段数1と高速である，

という特長がある．一方，スイッチ数が PE 数の2乗に比例して増えるという欠点があり，100 台以上の大規模マルチプロセッサでは実現が困難であると従来思われていた．しかしながら，最近の VLSI 技術および高密度実装技術の進歩に伴い，大規模なクロスバー網の実現も不可能ではなくなってきた．本システムでは 128×128 クロスバー網を実現するにあたり，図3に示すように，これを 256 (=16×16) 個の 8×8 クロスバー LSI に平面分割し，列方向 16 個および行方向 16 個をそれぞれバス接続するようにした．クロスバー網の機能の詳細は，3章で述べる．

2.3 メモリ

マルチプロセッサ・システムにおけるメモリアーキテクチャを特徴づける要素としては，

① メモリ配置形態: 集中メモリ構成か分散メモリ構成か．すなわち，メモリがグローバルメモリとして存在するか，あるいはローカルメモリとして存在するか．

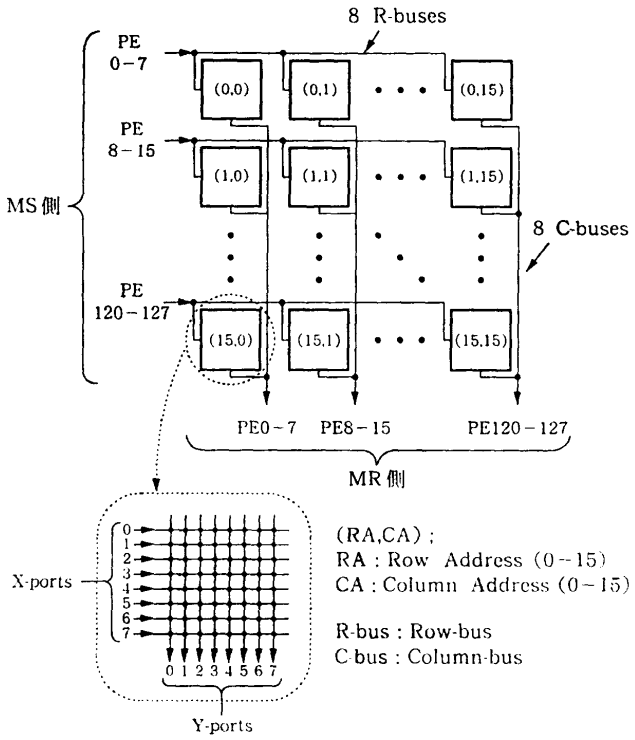


図 3 128×128 クロスバー網の構成
Fig. 3 Arrangement for 128×128 crossbar network.

② メモリ所有形態：あるメモリから見て，それが単一プロセッサに専有されているか，あるいは複数プロセッサに共有されているか．

③ プロセッサ間結合形態：密結合型マルチプロセッサか疎結合型マルチプロセッサか．すなわち，ある2個以上のプロセッサが共有メモリを介して結合されているか，そうでないか．

などがある[2),4)]．本システムでは，128 台の各 PE に 4 M バイトのローカルメモリを持たせる分散メモリ構成を採る．つまり，上記①に関しては固定構造である．しかしながら，プログラムにとって意味がある要素は上記の②および③，すなわち，プロセスあるいはプロセッサから直接アクセス可能な記憶空間の構成である．そこで，本システムでは，各ローカルメモリを自 PE に単に専有させるのではなく，他 PE からもクロスバー網を経由してアドレスにより直接アクセス（リモートメモリ・アクセスと呼ぶ）可能とする“ローカル／リモート・アーキテクチャ”を採用している．

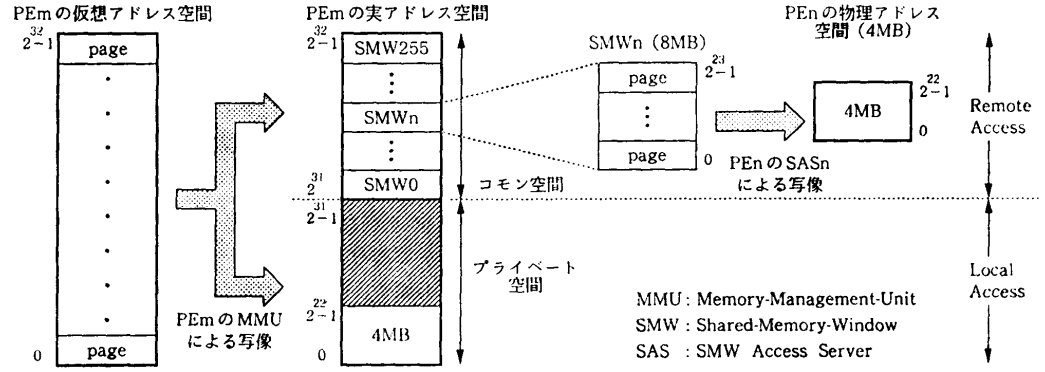
ローカル／リモート・アーキテクチャの実現方法として2，3の方式が提案されているが[8),9)]，本システムでは後述する理由により，仮想／実／物理の3種類のアドレスを導入した2レベル・アドレス変換によりこれを実現している．このアドレス変換過程は，以下のようになる（図4参照）．

（1） 仮想アドレス→実アドレス変換

SPARC マイクロプロセッサが出力する 32 ビット仮想アドレスを MMU のページング機構により 32 ビット実アドレスに変換する．このとき，4G バイトの実アドレス空間を次のように下位と上位の2個の空間に分割し，実アドレスがいずれの空間を指定しているかでローカルメモリ・アクセスかリモートメモリ・アクセスかを判定する．

① プライベート空間：下位 2G バイトの空間であり，それぞれの PE の私有領域である．実アドレスがプライベート空間を指定していれば第2レベルのアドレス変換を経ずに，自ローカルメモリにアクセスする．

② コモン空間：上位 2G バイトの空間であり，全 PE の共通領域である．コモン空間は，各 PE 対



ⓐ 仮想アドレス空間から実アドレス空間への写像　　ⓑ 実アドレス空間から物理アドレス空間への写像

図 4 アドレス空間の写像
Fig. 4 Addressing scheme.

応に 256 個の共有メモリウィンドウ (SMW) に分割される (現在 SMW 128～SMW 255 は未使用). この SMW にアクセス (SMW アクセスと呼ぶ) しようとすると, SMW に対応する他 PE のローカルメモリ (これをリモートメモリと呼ぶ) へクロスバー網を介してリモートメモリ・アクセスすることになる.

(2) 実アドレス→物理アドレス変換

SMW アクセスが起動されると, 当該 SMW を提供する PE において, SAS が実アドレスをページングにより物理アドレスに変換してメモリアクセスを遂行する.

このようにローカル／リモート・アーキテクチャでは, ローカルメモリおよびリモートメモリへのアクセスがプロセッサから見て全く同一であり, 唯一異なるのはアクセス時間のみとなる. 本アーキテクチャを採用するシステムには本システム以外に, 米 CMU の Cm*[8], 米 IBM ワトソン研の RP3[9], 米 BBN 社の Butterfly などがあり, いずれもそのアドレッシング機構が異なる. 本システムでは, アドレス変換過程を2レベルに分け, 第2レベルの (すなわち, ある SMW アクセスに関する) アドレス変換テーブルを当該 SMW を提供する PE で一元的に管理している点に特長がある. これは, OS による共有メモリすなわちコモン空間の管理を容易にすることを目的とする. つまり, SMW ページの置換および再配置を当該 SMW を提供する PE において独立に, かつ, 第1レベルのアドレス変換テーブルのいずれにも変更を与えることなしに行うことを可能とするためである[5].

また, SMW ページごとにアクセス可／不可を示すビットマップを MMU 内に設けている. このビットマップをプログラムすることで, プロセッサからアドレス指定で直接アクセス可能なコモン空間として様々な形態を構成できる. これにより, 任意の PE 群に関して, メモリ共有型密結合マルチプロセッサ, メッセージ交換型疎結合マルチプロセッサ, あるいは両者混合型マルチプロセッサ構成とすることが可能となる.

2.4 PE 間メッセージ通信機構

本システムにおける PE 間通信には, 高スループット, 低遅延などの通信本来の要件に加えて,

① 任意の PE 間接続形態 (トポロジー) の実現

② 密結合型および疎結合型双方の PE 間通信機能の提供

といった本システム特有の要件がある.

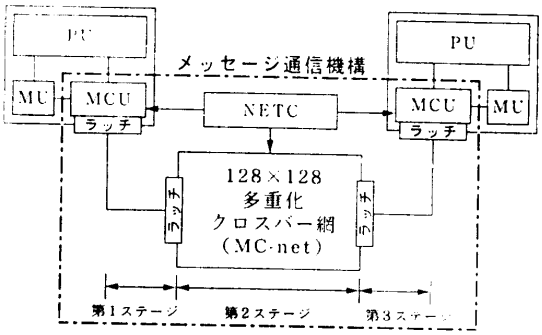


図 5 PE 間メッセージ通信機構
Fig. 5 Inter-PE communication subsystem.

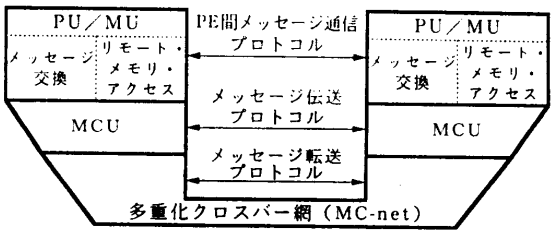


図 6 PE 間通信プロトコル
Fig. 6 Inter-PE communication protocol.

これらの要件に応えるため, 図5に示すように, 多重化クロスバー網 (MC-net: Multiplexed Crossbar network), ネットワークコントローラ (NETC)[3),6)], および, 各 PE のメッセージ通信ユニット (MCU) から成る PE 間メッセージ通信機構を備える. 本機構は3章で述べるように,

① クロスバー網の時間多重化

② パイプライン制御によるメッセージ転送

などの特長を有する.

本機構は, リモートメモリ・アクセスおよびプロセス間メッセージ交換といった PE 間で行われるすべての通信をつかさどるものであり, これら PE 間通信を以下の3階層プロトコル (図6参照) により実現する.

① メッセージ転送プロトコル: 最下層のプロトコルであり, 多重化クロスバー網上での回線接続／切断, データ転送, 方向切換えなどの手順を定める.

② メッセージ伝送プロトコル: 中間層のプロトコルであり, プリミティブメッセージ (4章参照) の組立, 送受, 解釈などの手順を定める.

③ PE 間メッセージ通信プロトコル: 最上位層のプロトコルであり, 次の2つに分かれる;

i) プロセス間メッセージ交換プロトコル: 一般には, PU 上のプログラム (OS の通信ハンド

ラなど）により定まる．

ii）リモートメモリアクセス・プロトコル：ローカルメモリにアクセスする際のプロトコルと同様である．

3. 多重化クロスバー網 (MC-net)

3.1 クロスバー LSI

表1に示す諸元を有する 8×8 クロスバー LSI を 256 個用いて，128×128 の回線交換方式のクロスバー網を構成する．個々のクロスバー LSI は，1つのポートに対する複数の回線接続要求の競合をどの時点で調停するかで，以下の2種類の調停機能を提供する．

① デマンドモード：プログラム実行時に回線接続要求が生じた際に動的に調停を行う．

② プリセットモード：プログラム実行前にソフトウェアによりあらかじめ調停を済ませ，その結果として得られた接続パターンを LSI に格納しておく．実行時には，LSI はその接続パターンに従って回線を単に設定するだけでよい．

3.2 多重化クロスバー網の動作モード

個々のクロスバー LSI の動作モード（デマンド／プリセット・モード）の組合せにより，クロスバー網全体としては次の3つの動作モードが可能である．

（1） 単一デマンドモード

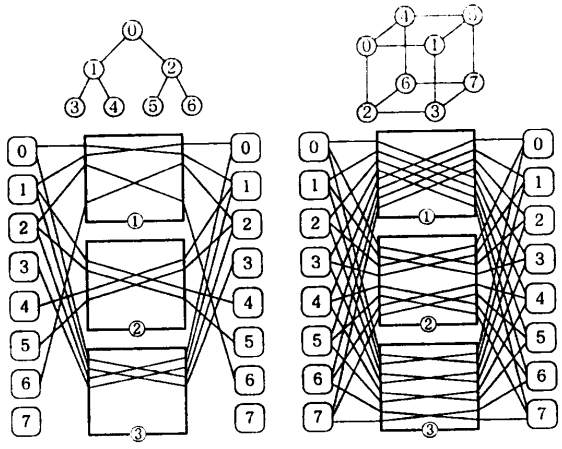
256 個のすべてのクロスバー LSI がデマンドモードで動作する．各 LSI および各 MR（メッセージレシーバ）は，MS（メッセージセンダ）からの回線接続要求に応えて動的に回線を接続する．このとき，2段階の調停，すなわち，LSI 自身による LSI 内調停と MR による LSI 間調停を同時に行う[6]．本モードではクロスバー網の動的網としての本来の能力を十分に発揮でき，通信パターンが実行時にしか決定されないような非定型的な応用分野に有効である．

（2） 単一プリセットモード

256 個のすべてのクロスバー LSI がプリセットモードで動作する．本モードでは，クロスバー網全体における PE 間接続形態が静的に一意に定まる．このとき，クロスバー網の一時における PE 間接続次数は高々1であるため，接続次数が2以上の PE 間接続形態を実現することができない．この問題の対処方法として，クロスバー網の多重化あるいは通信のルーティングなどが考えられるが，本システムではクロスバー網の多重化を採る．多重化の方法にはさらに，空間多重化と時間多重化の2方式があるが，現時点では時間多重化のみを採用し，回線接続時間を時分割するようにしている．このように時間多重化していることから，本クロスバー網を多重化クロスバー網と呼ぶ．

表1 クロスバー LSI の諸元
Table 1 Specifications of crossbar LSI.

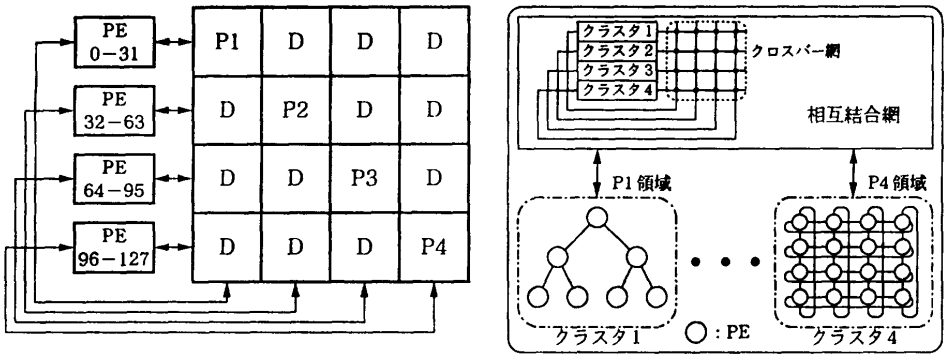
交換数	8×8
交換方式	回線交換
転送方向	双方向（半二重）
転送データ幅	ポート当り1バイト
調停方式（デマンド・モード時）	優先順位選択方式の多入力非同期調停方式
放送機能	サポート
動作周波数	16.7 MHz
パッケージ	223 pin PGA パッケージ
プロセス技術	1.5 μmCMOS



（a）2進木の実現例 （b）2進3キューブの実現例
①, ②, ③：制御メモリに記憶された結合パターン

図7 PE 間接続形態の実現例
Fig. 7 Embedment of topologies of inter-PE connection.

本モードにおいては，まずプログラム実行前に，PE 間接続形態を時系列として各クロスバー LSI および各 MCU 内の制御メモリに格納しておく[6]．実行時には，このメモリの内容を順次読み出し，その内容に従って回線接続を行う．時分割の高速化のため，クロスバー LSI の制御メモリには 16 パターンまでの接続形態を格納できるようにしている（制御メモリの内容は随時書換え可能であり，16 パターン以上の接続形態も当然実現できる）．**図7**に，2種類の PE 間接続形態（2進木と2進3キューブ）の本モードでの実現例を示す．この例では接続次数すなわち時間多重度



D：デマンド・モード領域　P：プリセット・モード領域
（a） MC-net の領域分割の例　（b） 各領域の動作形態の例

図 8 ハイブリッドモードの使用例
Fig. 8 Example of hybrid-mode operation.

はいずれも3であり，図に示すように，制御メモリに格納された3パターンを順次切り換えながら回線接続時間を3分割する．

このように，本モードでは，PE 間接続形態が実行前に一意に定まることから実行時に競合が生じることがなく，したがって，調停ないし回線待ちのオーバヘッドが入らない．本システム以外にも，本モードに類似した相互結合網を用いたシステムとして，米 IBM ワトソン研の GF-11[10] や西独 Parsytec 社の Megaframe Supercluster[11] などがある．本モードは，大部分の科学技術計算 (PDE, FFT, QCD など) に代表されるような通信形態が定型的な応用分野に対して有効である．

（3） ハイブリッドモード

個々のクロスバー LSI が，デマンドモードないしプリセットモードのいずれかで動作する．したがって，クロスバー網上に 8 PE 単位で任意のクラスタ構成を形成できる．**図 8** にハイブリッドモードの使用例を示す．図では，木構造，トーラス構造のクラスタがクロスバー網で結合される様子を示している．

3.3 パイプライン制御によるメッセージ転送

1つのクロスバー網上で，高スループットを要求するプロセス間メッセージ交換，および，低遅延を要求するリモートメモリ・アクセス，といった性質の異なる PE 間通信を実現しなければならない．そこで，本システムではクロスバー網上のメッセージ転送方式として，パイプライン制御に基づくクロック同期転送を採用してこれらの要求に応えている．

この方式は PE 間の通信路を遅延要素ごとに分割して，これを1つのパイプラインステージと見なし，ステージ間のラッチの制御を単一クロックに同期させて行う方式である．このとき，パイプラインピッチは各ステージの遅延のうち最大のもので決まる．図5に示すように，通信路を3ステージに分割し，3段のパイプライン構成としている．パイプラインピッチを 60 ns に設定し，最大転送速度 16.7 M バイト／秒，網内遅延 180 ns を実現している．

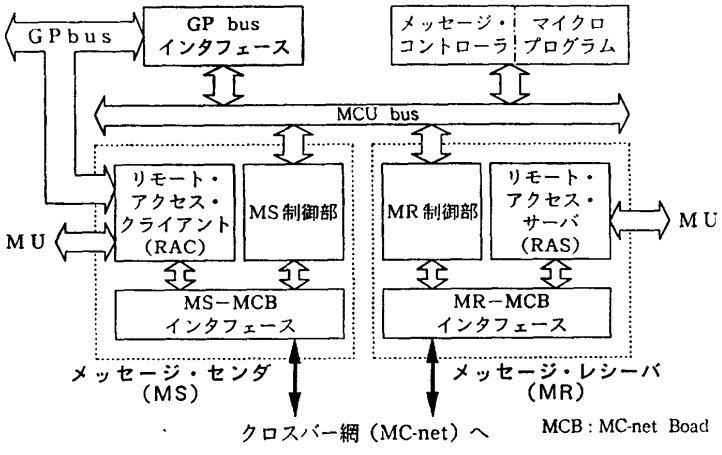


図 9 メッセージ通信ユニットの構成
Fig. 9 Block diagram of MCU.

4. メッセージ通信ユニット (MCU)

PE 間通信処理をプロセッサユニット (PU) と並列かつ独立に行うため，通信処理を専用に行うメッセージ通信ユニット (MCU) を各 PE に装備する．MCU は**図 9** に示すように，メッセージセンダ

(MS), メッセージレシーバ (MR), および, メッセージコントローラで構成され, おのおの独立に動作可能である. MCUは, PU 上のプログラムに対して PE 間通信プロトコル (図6参照) のどの階層を見せるかで, 図 10 に示すように,

① I/O デバイス・インタフェース

② リモートメモリ・インタフェース

③ メッセージ・コプロセッサ・インタフェース

といった3種類のインタフェースを提供する. 以下, これらインタフェースごとに, MCU の主要機能を述べる.

4.1 I/O デバイス・インタフェース

PE 間通信プロトコルの中間層のメッセージ伝送プロトコルを直接 PU 上のプログラムに見せた場合, MCU はあたかも低機能かつ高速な I/O デバイスのように振舞う.

メッセージ伝送プロトコルにおいては, 表2に示すプリミティブメッセージが定義されている. 本インタフェースでは, これらプリミティブメッセージを用いることで, プログラム (たとえば, OS の通信ハンドラなど) が任意形式のプロセス間メッセージ交換を実現することを可能としている. このときでも, 最下層のメッセージ転送プロトコルは隠蔽されるので, PU 上のプログラムは, 多重化クロスバー網の動作モードや転送手順などを意識する必要はない.

表 2 プリミティブ・メッセージ
Table 2 Primitive messages.

メッセージの種類	メッセージの意味
プロセス間メッセージ†	プロセス間通信のためのメッセージ (メッセージ長2~508バイト)
割込み要求メッセージ†	他 PE の PU へ外部割込みを行うメッセージ
制御メッセージ†	他 PE を停止, リセット, および再スタートさせるメッセージ
読出し/書込み‡	他 PE のローカル・メモリへリモート・アクセスを行う
ラインフェッチ‡	
不可分読出し書込み‡	
リモート・ページテーブル・アクセス‡	他 PE のローカル・メモリ上のページテーブルへリモート・アクセスを行う
キャッシュパージ‡	他 PE のキャッシュパージを行う
応答メッセージ‡	リモート・メモリ・アクセスに対する応答メッセージ

† OS が明示的に伝送を依頼するメッセージ (ヘッダ長4バイト)

‡ リモート・メモリ・アクセスに伴い MCU で暗黙的に作成され伝送されるメッセージ (ヘッダ長6バイト. アドレスはヘッダに含まれる. ただし, 応答メッセージはヘッダを持たない.)

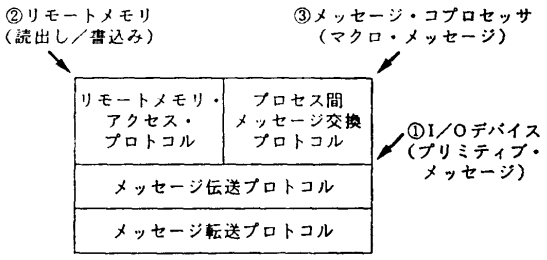


図 10 MCU が提供するインタフェース
Fig. 10 MCU interfaces.

PU 上のプログラムは, MS および MR 制御部内の送信/受信バッファに対してプリミティブメッセージの書込み/読出しを行うことで, プリミティブメッセージの伝送を行う. このプログラムによるバッファ操作とメッセージ転送のためのバッファ操作との衝突を回避する目的で, 送信/受信バッファはデュアルポート構成とした. 具体的には, 送信バッファには高速 FIFO メモリを用いて, クロック同期転送方式に従った間断ないメッセージ送信を可能としている. また, 受信バッファには大容量デュアルポート SRAM を用いて, バッファオーバフローによるメッセージ転送の停滞を避けている.

4.2 リモートメモリ・インタフェース

PE 間通信プロトコルの最上位層のリモートメモリアクセス・プロトコルを PU から見た場合, MCU はあたかもリモートメモリのように振舞う.

このとき, 2章で述べたように, ローカルメモリ・アクセスもリモートメモリ・アクセスも, そのプロトコルはまったく同一である必要がある. すなわち, PU から見た場合, リモートメモリもローカルメモリと同様な手順で, 命令フェッチや LOAD/STORE 命令によるデータの読出し/書込みができなければならない. よって, リモートメモリアクセス・プロトコルからメッセージ伝送プロトコルへのプロトコル変換が必要となる.

このプロトコル変換に携わるのが, リモートアクセス・クライアント (RAC) およびリモートアクセス・サーバ (RAS) である. RAC は, PU から出されたメモリアクセス要求をプリミティブメッセージに変換して自動的に多重化クロスバー網に送出する. ただし, 当該リモートメモリを提供する PE との回線接続が不可能な場合 (当該 PE が別の回線に接続されている場合や, 単一プリセットモード動作時で当該 PE との回

線が存在しない場合など）には，メモリ例外としてその旨を PU に知らせる．一方アクセスされる側の RAS は，受信したプリミティブメッセージの解釈を行いメモリアクセスを遂行し，アクセス結果をアクセスした側の RAC に返送する．このときのリモートメモリへのアクセス時間はプロトコル変換時間に大きく依存することから，RAC および RAS は布線論理により構成している．

4.3 メッセージ・コプロセッサ・インタフェース

PE 間通信プロトコルの最上位層のプロセス間メッセージ交換プロトコルを PU 上のプログラムから見た場合，MCU はあたかもプロセス間メッセージ交換機能を備えたメッセージ・コプロセッサのように振舞う．

一般の疎結合型マルチプロセッサにおいては，OS の通信ハンドラによってプロセス間メッセージ交換機能が提供されており，そのプロトコルも通信ハンドラにより定まる．本システムでも 4.1 節で述べたように，MCU のインタフェースとして I/O デバイス・インタフェースを用いて，通信ハンドラにより任意のプロセス間メッセージ交換機能を提供することができる．しかし，アプリケーションプログラムと通信ハンドラが同一プロセッサ上で動作することから，プロセス間メッセージ交換に伴う通信ハンドラの処理がオーバヘッドとなる可能性がある．このオーバヘッドを軽減するために導入されるのが，プロセッサと並行動作可能で，通信ハンドラの機能を一部オフロードしたメッセージ・コプロセッサである[12]．

本インタフェースにおけるメッセージ・コプロセッサの役割を担うのが，メッセージコントローラである．本コントローラには，高速かつ柔軟な処理を可能とするため，マイクロプログラム制御の 1 チップ・マイクロコントローラ（WSI 社製 PAC 1000[13]）を採用している．そのマイクロアーキテクチャは，64 ビット幅水平型マイクロ命令，16 ビットシーケンサ，16 ビット ALU，制御記憶容量 1 k 語となっており，動作周波数は PU と同じく 16.7 MHz である．

このときの PU 上のプログラムとメッセージ・コプロセッサとの間のインタフェースは，プリミティブメッセージより高レベルのメッセージ（これをマクロメッセージと呼ぶ）となり，その機能はマイクロプログラムにより柔軟に設定することが可能である．

5. 性能予測

単一プリセットモードにおける PE 間メッセージ通信機構の性能予測を行った．

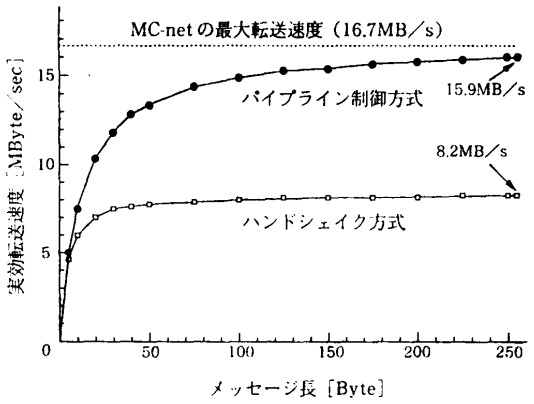


図 11 プロセス間メッセージ伝送時の実効転送速度
Fig. 11 Effective bandwidth for interprocess message transmission.

表 3 メモリ・アクセス時間
Table 3 Memory access time.

タイプ ＼ アクセス先 ／ サイズ		キャッシュ	ローカルメモリ	リモートメモリ	
読出し	バイト(1B)	60ns	180ns	1440ns	
	半語 (2B)			1500ns	
	語 (4B)			1620ns	
	倍長語(8B)	120ns	240ns	1860ns	
書込み	バイト(1B)	—	180ns	1260ns†	180ns‡
	半語 (2B)			1320ns†	
	語 (4B)			1440ns†	
	倍長語(8B)		240ns	1740ns†	240ns‡
ラインフェッチ	ライン(32B)	—	600ns	3300ns	

†: 同期書込み　‡: 非同期書込み

図 11 は，プリミティブメッセージの 1 つであるプロセス間メッセージを，パイプライン制御によるクロック同期転送（3.3 節参照）した際の実効転送速度を示す．ここでは，受信バッファのオーバフローは生じないことを仮定している．クロスバー網自身の最大転送速度 16.7 M バイト／秒に対して，256 バイト長メッセージ伝送時の実効転送速度が 15.9 M バイト／秒となり約 95% の転送効率を達成している．参考のために，当初採用を検討した，一般によく用いられているハンドシェイク制御によるクロック非同期転送を適用した場合の実効転送速度を図中に示した．同一条件での実効転送速度が 8.2 M バイト／秒であり，約半分の性能しか得られない．

表 3 はメモリアクセスにおける，アクセスサイズと

アクセス時間の関係を示している．ただし，メモリアクセスに際しPE内部でのバス競合は起こらないと仮定している．4バイト読出し時におけるキャッシュメモリ (CM)，ローカルメモリ (LM) およびリモートメモリ (RM) のアクセス時間比は，

$$CM : LM : RM = 1 : 3 : 27 \quad (1)$$

となる．また，キャッシュ・ミスヒット時のラインフェッチにおけるアクセス時間比は

$$LM : RM = 1 : 5.5 \quad (2)$$

となる．ちなみに，米IBMワトソン研のRP3では，CM, LMおよびRMのアクセス時間比を

$$CM : LM : RM = 1 : 10 : 15 \quad (3)$$

と設定している[14]．本システムのメモリアクセス時間をRP3のそれと比較すると，RMは約2倍と低速だが，LMは約1/3と高速である．ローカル/リモート・アーキテクチャにおける適切なLM:RMアクセス時間比は，応用問題，データ配置アルゴリズムなどに依存して簡単には求められないが，今後の検討が必要である．

6. おわりに

以上，可変構造型並列計算機のPE間メッセージ通信機構について，構成，機能および予測性能を述べた．多重化クロスバー網は1つのクロスバー網の時間多重化を図ったものであり，その3種の動作モードにより任意のPE間接続形態を実現可能としている．また，MCUはPUに並行してPE間通信処理をつかさどり，1つのクロスバー網上でプロセス間メッセージ交換およびリモートメモリ・アクセスといった性質の異なるPE間通信を提供する．これにより，本機構は，プロセッサ-プロセッサ結合およびプロセッサ-メモリ結合として任意の結合形態をプログラムに対し提供可能である．また，その性能は，クロスバー網自身の転送能力を十分活かすものと予測している．

現在，ハードウェアの論理設計はほぼ終了している．ハードウェアの開発と並行して，並列/分散OS，および，並列プログラミング言語を含んだ並列プログラミング環境の構築を進めている[15),16)]．本システムの詳細な性能評価は，ハードウェアの完成を待って，さらに並列/分散OSおよび並列プログラミング環境をも含めた実機上での総合的な評価を予定しており，これらの報告については別の機会に譲りたい．

謝辞 本プロジェクトの初期段階で開発に携わっていた田中幸二氏（現在 日本電気(株)），安富伸浩氏（現在 富士ゼロックス(株)），現在我々とともに設計・開発を行っている福澤祐二，廣谷良彰，岩田英次，甲斐康司，草野和寛，恒冨邦彦の各氏，および，日頃ご討論いただく富田研究室の皆様に感謝いたします．

クロスバーLSI開発においては(株)東芝・総合研究所・情報システム研究所の小柳滋博士，田邊昇氏，PE開発においては富士通(株)本体事業部・スーパーコンピュータ開発部の内田啓一郎氏，高村守幸氏，京セラ(株) LSIデザイン事業部の重村慎二氏，システム実装設計に関してはアジアエレクトロニクス(株)の作田信彦氏ならびにシステム機器事業部，ニシム電子工業(株)の和田勲夫氏に，ご助言ご協力をいただいている．謹んで感謝いたします．

なお，本研究は一部文部省科研費一般研究による．

参考文献

1) 富田眞治：並列計算機構成論，昭晃堂 (1986).
2) 村上ほか：可変構造型並列計算機のシステム・アーキテクチャ，情報処理学会コンピュータ・アーキテクチャシンポジウム論文集，pp. 165-174 (1988).
3) 森ほか：可変構造型並列計算機のPE間メッセージ通信機構，情報処理学会並列処理シンポジウム JSPP '89 論文集，pp. 123-130 (1989).
4) Murakami, K. et al.: The Kyushu University Reconfigurable Parallel Processor—Design Philosophy and Architecture—, *Proc. IFIP 11-th World Computer Congress*, pp. 995-1000 (1989).
5) Murakami, K. et al.: The Kyushu University Reconfigurable Parallel Processor—Design of Memory and Intercommunication Architectures—, *Proc. 1989 Int. Conf. Supercomputing*, pp. 351-360 (1989).
6) 蒲池ほか：可変構造型並列計算機のネットワーク制御方式，信学技法，CPSY 89-16 (1989).
7) Vick, C. R. et al.: Adaptable Architectures for Supersystems, *Computer*, Vol. 13, No. 11, pp. 17-35 (1980).
8) Gefinger, E. F. et al.: *Parallel Processing: The Cm* Experience*, Digital Press (1987).
9) Brantley, W. C. et al.: RP 3 Processor-Memory Elememt, *Proc. 1985 Int. Conf. Parallel Processing*, pp. 782-789 (1985).
10) Beetem, J. et al.: The GF-11 Super Computer, *Proc. 12th Int. Symp. Computer Architecture*, pp. 108-115 (1985).
11) Kübler, F. D. et al.: A Cluster-Oriented Architecture for the Mapping of Parallel Processor Networks to High-Performance Appli-

cations, *Proc. 1988 Int. Conf. Supercomputing*, pp. 179-189 (1988).
12) Peterson, J. et al.: A High-Speed Message-Driven Communication Architecture, *Proc. 1988 Int. Conf. Supercomputing*, pp. 355-366 (1988).
13) Waferscale Integration, Inc.: *High-Performance Programmable Standalone Microcontroller (PAC)*, Waferscale Integration, Inc. (1988).
14) Pfister, G.F. et al.: The IBM Research Parallel Processor Prototype (RP 3): Introduction and Architecture, *Proc. 1985 Int. Conf. Parallel Processing*, pp. 764-771 (1985).
15) 福澤ほか：可変構造型並列計算機のソフトウェア，第3回情報処理学会九州支部研究会報告，pp. 29-38 (1989).
16) 福田ほか：可変構造型並列計算機の並列/分散オペレーティング・システム，情報処理学会オペレーティング・システム研究会資料，OS-43-8 (1989).

(平成元年5月30日受付)
(平成元年9月12日採録)

森 眞一郎（正会員）

1963年生．1987年熊本大学工学部電子工学科卒業．1989年九州大学大学院総合理工学研究科情報システム学専攻修士課程修了．現在同大学院博士課程に在学中．並列処理，計算機アーキテクチャの研究に従事．

蒲池 恒彦（学生会員）

1964年生．1988年九州大学工学部情報工学科卒業．現在，同大学大学院総合理工学研究科情報システム学専攻修士課程在学中．並列処理，計算機アーキテクチャの研究に従事．

濱口 一正（正会員）

昭和38年生．昭和61年，福岡大学工学部電気工学科卒業．平成元年，九州大学大学院総合理工学研究科情報システム学専攻修士課程修了．同年，キャノン(株)入社．現在，同社情報システム研究所コンピュータシステム研究部に所属．計算機アーキテクチャ，並列処理に興味をもつ．

村上 和彰（正会員）

1960年生．1982年京都大学工学部情報工学科卒業．1984年同大学院工学研究科情報工学専攻修士課程修了．同年富士通(株)本体事業部に入社．主として汎用計算機Mシリーズのアーキテクチャ開発に従事．1987年九州大学工学部助手，1988年同大学院総合理工学研究科情報システム学専攻助手，現在に至る．計算機アーキテクチャ，スーパーコンピューティング，並列処理等の研究に従事．著書「計算機システム工学（共著，昭晃堂）」．電子情報通信学会，ACM, IEEE-CS 各会員．

福田 晃（正会員）

1954年生．1977年九州大学工学部情報工学科卒業．1979年同大学院修士課程修了．同年 NTT 研究所入所．1983年九州大学大学院総合理工学研究科情報システム学専攻助手．1989年同大学助教授，現在に至る．工学博士．計算機システムの性能評価，並列処理，オペレーティング・システムなどに興味をもつ．訳書：オペレーティングシステムの概念（共訳，培風館）．電子情報通信学会，日本 OR 学会，ACM, IEEE 各会員．

末吉 敏則（正会員）

昭和28年生．昭和51年九州大学工学部情報工学科卒業．昭和53年同大学院修士課程修了．同年九州大学工学部情報工学科助手．その後，九州大学大学院総合理工学研究科助教授を経て，平成元年4月九州工業大学情報工学部知能情報工学科助教授．工学博士．並列処理システム，分散処理システムの研究に従事．電子情報通信学会，IEEE 各会員．

富田 眞治（正会員）

1945年生．1968年京都大学工学部電子工学科卒業．1973年同大学院博士課程修了．この間，零交さ波による音声合成の研究に従事．工学博士．同年京都大学工学部情報工学教室助手．1978年同助教授．1986年九州大学大学院総合理工学研究科教授，現在に至る．計算機アーキテクチャ，並列処理システムなどに興味を持つ．著書「並列計算機構成論」「計算機システム工学」「並列処理マシン」など．電子情報通信学会，IEEE，ACM 各会員．