7T-1　多段結合ネットワークを用いた共有メモリ型マルチマイクロプロセッサの設計

田胡　和哉、森下　巖、中島　敦、森丘　力（東京大学　工学部　計数工学科）

1．まえがき

共有メモリを用いた並列処理計算機では、共有メモリへのアクセスを効率良く行なえるようにすることが重要である。共有メモリへのアクセスの競合を軽減する一つの方式として、共有メモリを多数の単位に分割し、プロセッサとメモリをネットワークを用いて結合する方式が考案されている。メモリ・アクセス要求をパケットとして実現し、２入力２出力のノードを単位とする多段構造のネットワークを用いてパケットを伝達することにより、比較的少数の部品を用いてこのようなネットワークを実現することができる[1,2]。その一方、パケットの伝送による遅延に配慮してシステムの設計を行なう必要がある。これまでに、68000プロセッサを用いた多段結合ネットワーク・システムの設計を行ない、この遅延を考慮しても実用性のあるシステムが実現できる可能性が高いことを確認した。

2．実現方式

図１に、多段結合ネットワーク・システムの構造を示す。システムは、同数のプロセッサ・エレメント(PE)とメモリ・

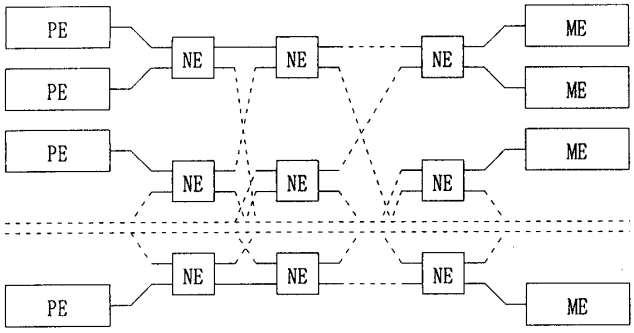

図1　多段結合ネットワークを用いたシステムの構造

エレメント(ME)を、結合路に冗長性のない多段結合ネットワークであるオメガ・ネットワークを用いて結合することによって構成されている。MEの集合は、全体で一つの共有メモリを構成している。PEは、共有メモリへのアクセスごとに、アドレス、データ、メモリへの操作の識別子等を含むパケットを作成する。パケットは、ネットワークを経由してアドレス語の値に対応するMEに伝達され、その返答は、逆方向の経路を経由してPEに送られる。

オメガ・ネットワークを構成するネットワーク・エレメント(NE)は、入力パケット中のアドレス値のいずれかのビット位置を観測し、その値に対応して２つの出力のいずれかからそのパケットを送出する。また、２つの入力要求の調停およびバッファリングを行う。Ｎ個のPEおよびMEからなるシステムでは、(log N)段のNEを結合し、それぞれの段のNEにおいてアドレス語の異なるビット位置を観測することにより、Ｎ対Ｎ結合を実現することができる。

オメガ・ネットワークを用いることにより、N(log N)個の部品でネットワークを実現することができる。また、経路の決定を各ノードで局所的に行なうことが可能であり、ハードウェアの実現が容易である。ネットワークを経由するメモリ・アクセスでは、MEのスループットとほぼ同一のスループットが得られる。一方、個々のアクセスの返答には、各ノードにおけるパケットの授受に要する時間の２(LOG N)倍の遅延が生じる。

3．ハードウェアの設計

3．1　全体の構造

通常のPEは、プロセッサ、局所メモリ、および、キャッシュ・メモリから構成される。PEのうち一つには、ホスト計算機との結合に用いるDMA転送の制御機構、および、周辺機器を付加する。このPEを、サーバPEとよぶ。NEは、図２に示すような、パケットを保持するバッファ・ラッチ、および、制御

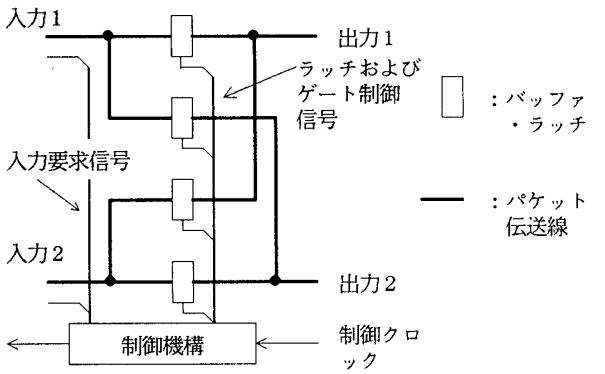

図2　NEの構造

機構からなる回路を２組用いて実現される。MEは、ネットワークとのインタフェース機構、および、通常のメモリ素子から構成される。

表１に、パケットのデータ構造を示す。UAは、アドレス値のうちMEを識別する部分であり、LAは、各ME内のアドレスを識別する部分である。OPはメモリに対する操作、PEIDは、PEの識別子を表す。DATAは、MEへの書込み、および、MEからの読み出しデータを保持する。

通常のPE、ME、および、NEは、単一のボードを用いて実現する。サーバPEは、VMEバス・モジュールを組合せて実現する。エレメント間は、一つのパケットを並列に伝送する多芯ケーブルを用いて結合する。

表1　パケットのデータ構造

ﾌｨｰﾙﾄﾞ名	ﾋﾞｯﾄ数
UA	8
LA	20
OP	4
PEID	8
DATA	16

3．2　共有メモリへのアクセス

(1)　アドレス空間の割当て

各PEの物理アドレス空間は、局所メモリ領域、共有メモリ領域、および、同期領域から構成される。局所メモリ領域には、PE上に実装されているROM、および、RAMが割当てられる。共有メモリには、キャッシュ機構を経由してアクセスが行なわれる。プロセッサ間で共有される共有データを、共有メモリ領域に配置する。同期領域は、プロセッサ間の同期を実現するために用いられる。

(2)　キャッシュ機構

キャッシュ・メモリは、直接マッピング方式を用いて実現する。キャッシュ・メモリ上に配置されるデータが限定されているために、キャッシュ・メモリの割当て制御の乱れは生じにくい。キャッシュ機構は、共有データのバッファリングを行なうとともに、ネットワークとのインタフェースを実現する。

共有メモリ領域からの読み込みにおいてキャッシュ・ミスが生じた場合には、読み出し要求のパケットをネットワーク

Design of a Shared Memory Multiprocessor System Using Multistage Interconnection Network
Kazuya TAGO, Iwao MORISHITA, Atushi NAKAJIMA and Tsutomu MORIOKA
University of Tokyo

に送出した後、その返答のパケットを待合わせる。待合わせの間、プロセッサは停止する。共有メモリへの書込みは、ライト・スルー方式によって実現する。書込みの終了の待合わせは行なわない。

(3) プリフェッチ

ネットワークの特性により、キャッシュ・ミスによる共有メモリへのアクセスの待合わせに要する時間が比較的大きい。そこで、キャッシュ・ミスの機会を減らすために、ソフトウェアがキャッシュ・メモリへのデータ読み込みを示唆できるようにする。プリフェッチ命令

```
prefetch addr,size
```

により、共有メモリ上のaddrからsize分の領域がMEからキャッシュ・メモリに転送される。実際に共有データ利用する以前にこの命令を発行するように、ソフトウェアを実現する。

(4) NEの制御

NEは、同期式の調停回路を用いて制御する。後段のNEから前段のNEにクロックを伝達し、このクロックにあわせて転送の制御を行なうことにより、1段のNEあたり1クロックの遅延でパケットを転送することができる。クロック信号として、20 MHzの信号を用いる予定である。

3．3 プロセッサ間の通信および同期

(1) 共有メモリとキャッシュ・メモリの内容の整合性の実現

共有メモリへのアクセスでは、パケットが、生成された順番のままMEに到着するとは限らない。また、返答に遅延が生ずる。そこで、たとえば、プログラムが共有メモリ上の際どい領域を終了するためには、以前発行した際どい領域への書込み要求がすべてMEに到達していることを確認する必要がある。これを実現するために、共有メモリへのアクセスを行なうと1増加し、その返答が帰ると1減少するカウンタをPEに設ける。これを、アクセス・カウンタ(AC：Access Counter)とよぶ。ACが0のときに、キャッシュ・メモリの内容と、対応する共有メモリの内容が一致している。

(2) プロセッサ間の同期および通信

プロセッサ間の同期および通信を、反射機構を用いて実現する。パケットのOPフィールドが反射型であるものがMEに到着すると、DATAフィールドの値とPEIDフィールドの値を交換して返答のパケットを作成する。あるPEが同期領域にアクセスすると、いずれかのプロセッサからそのアドレスに対する反射パケットが到来するまでそのPEが停止する。反射パケットが到着すると、そのパケットのDATAフィールドの値が読み出される。これにより、プロセッサ間の通信を実現することができる。

(3) フェッチアンドアド

フェッチアンドアド命令(f_a命令)は、共有メモリからの読み出しと、同一メモリ・セルに対する加算をアトミックに実行する。

4． ソフトウェア

4．1 ベクトル化方式による並列処理

多段ネットワーク・システムの利用法の一つとして、手続きを単位とするベクトル化計算による方式を検討している。C言語をベクトル化計算向きに拡張して用いる。図3に、ベクトル化の方法の概要を示す。C言語の記述では、for文を用いてfunc手続きをN回繰り返して呼出す。これに対して、あらたに設けたvect文では、各プロセッサ上で引数の異なるfunc手続きをN個並列に実行する。key変数は、f_a命令を用いて並列に生成する。

図4に、このようなベクトル化演算を実現するために必要な変数領域の割当て法を示す。大域変数は、共有メモリ上に配置する。また、局所変数のうち、並列に実行される手続きの内部で参照されるものも、共有メモリ上に配置する。局所変数を共有メモリ上に配置することを、export宣言を用いて

```
int i;  ------------------------------------  大域変数
main()
{
    export int key;  ---------------------  局所変数

    vect(key=1; key<N; key++) ------------  vect文
        func(key); ---------------------  並列実行の
}                                          対象
```

図3 ベクトル化計算

指定する。局所変数の領域は、スタック構造をとる必要があるので、共有メモリ上にPEごとのスタック領域を割当てる。一方、式評価のための作業領域、および、通常の手続き呼出しを制御するために使用されるスタックは、局所メモリ領域上に配置する。

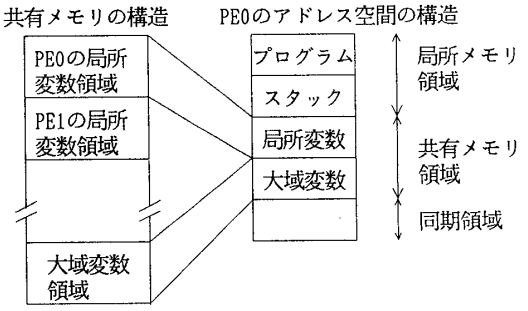

図4 変数領域の割当

vect文は、start命令、end命令、wait命令、および、f_a命令を用いて実現される。実行すべき仕事のないプロセッサは、同期領域を参照して停止している。start命令により、停止中のプロセッサを起動し、手続きの実行を依頼する。実行の終了は、wait命令によって待合わせる。start命令

```
start  N, addr, arg
```

は、ACの値が0になるのを待ちあわせ、反射パケットを生成し、N個のプロセッサにaddr、および、argを渡す。起動されたプロセッサは、addrから実行を開始する。arg番地に存在する引数列を局所メモリ領域に取込み、key変数を生成した後、実行すべき手続きを呼出す。手続きの実行が終了すると、end命令により、vect命令を発行したプロセッサに処理の終了を通知する。

4．2 オペレーティング・システム

通信で結合された軽量なプロセスの集合を用いて実現された、UNIXと同一仕様を持つ分散型オペレーティング・システムを用いて制御する。これにより、利用者プログラムから、UNIXのシステム・コールを利用することができる。システム・コールは、サーバPEに反射機構を用いて伝達する。

5． むすび

多段結合ネットワークを用いた共有メモリ型マルチプロセッサシステムの実現方式について述べた。ネットワーク1段あたり100n sec程度の遅延で伝達できる見込であり、キャッシュ機構を用いることにより、実用的な性能が得られると期待できる。今後、68030プロセッサを8個用いた試作システムの実現、および、その評価を行なう予定である。

参考文献

[1]Gottlieb,A. et.al : The NYU Ultracomputer-Designing and MIMD Shared Memory Parallel Computer, IEEE trans. on computers, Vol.c-32, No.2, pp.175-189 (1983).

[2]G.F.Pfister et al.:An Introduction to the IBM Research Parallel Processor Prototype(RP3), in J.J. Dongarra ed., Experimental Parallel Computing Architectures,pp.123-140, North-Holland, Amsterdam (1987).