計算機アーキテクチャ 59−5
オペレーティング・システム 29−5
(1985・10・24)

クラスタ方式バス結合型並列計算機

A Clustered Multiprocessor System Interconnected via Multiple Buses

中田 武男[+] 安倍 正人[++] 堀口 進[+] 川添 良幸[++] 重井 芳治[+]
Takeo Nakada Masato Abe Susumu Horiguchi Yoshiyuki Kawazoe Yoshiharu Shigei

[+] 東北大学工学部
Faculty of Engineering , Tohoku University

[++] 東北大学情報処理教育センター
Education Center for Information Processing, Tohoku University

1．はじめに

計算機システムを高速化する方法には、論理素子の性能向上によるものと、アーキテクチャによるものがある。これまでは、主に真空管に始まりトランジスタ、ＩＣ、ＬＳＩ、そしてＶＬＳＩに至る論理素子の高集積化が図られてきた。しかし、シリコンを用いた素子の性能向上は限界に近づいてきている。シリコンに代わってGaAsやジョセフソン素子などの論理素子も研究されてはいるが、シリコン素子と比較しての性能向上は１桁程度にとどまる。しかし、科学技術計算分野では、さらに高速の計算機が要求されている。

一方、演算処理を分割し同時に並列して処理を行う並列計算機の研究も1950年代頃から行われ、種々のアーキテクチャが提案され試作が行われている[1][2][3]。また、近年のＶＬＳＩ技術の急激な進歩により高性能なマイクロプロセッサを多数用いたマルチマイクロプロセッサシステムの実現が可能になり、Ｃｍ*[4]，ＰＡＸ[5]等実際に製作されてきている[6]。さらに、並列処理アルゴリズム[7][8]や並列処理記述言語の研究[9][10]も盛んに行われている．しかしながら、並列処理アルゴリズムが容易に記述できるソフトウェアと、作成したプログラムを実際に実行できるハードウェアを合わせ持ったシステムを実現した例は、それほど多くはない[3][5]。

マルチプロセッサシステムのプロセッサ結合方式は大別してバス結合型とネットワーク型の２方式になる。ネットワーク型のプロセッサ結合方式では、全てのプロセッサを互いに結合させる完全結合ネットワークを用いればプロセッサ間通信のためには理想的な結合形態となるが、プロセッサ数が多くなると配線が複雑になりすぎて実現することは困難になる。

バス結合方式の最もシンプルなシステムは単一バスに多数のプロセッサを接続した並列処理システムである。このシステムにおいてはプロセッサの処理量が少ない場合プロセッサ数が増すにつれ、プロセッサ間通信のためのバス競合によるオーバヘッドが増大し処理効率が低下してしまう。しかし、共有メモリとバスを用いたマルチプロセッサは、ハードウェア構造が簡単で、アルゴリズムに対して柔軟性があるため、並列処理計算機システムとして有効である。また、バス競合を緩和する方式としては、複線バス方式やプロセッサをいくつかのクラスタ単位にまとめるクラスタ方式がある。複線バス方式ではすべてのプロセッサ間通信が同様に行える点では有効であるが、ハードウェア量はかなり多くなる。クラスタ方式ではバスを階層化するため、異なるクラスタ内に属するプロセッサ間の通信は同一クラスタ内のプロセッサ間通信よりオーバヘッドが大きくなるが、ハードウェア量は複線バス方式よりは少なくてすむ。

我々は並列処理アルゴリズムや並列処理記述言語の開発を目的とするバス結合型マルチプロセッサシステムの実現のために、単一バスを用いたマスタ・スレーブ方式の並列処理システム

の効率検討と試作を行ってきた[11][12]。本稿では、クラスタ方式を用いたバス結合型並列計算機の概要を述べ、その試作機の処理効率の実測値および検討結果について報告する。さらに、クラスタ間を完全結合したシステムについて検討した結果を報告する。

2．クラスタ方式を用いた並列処理システムの基本動作

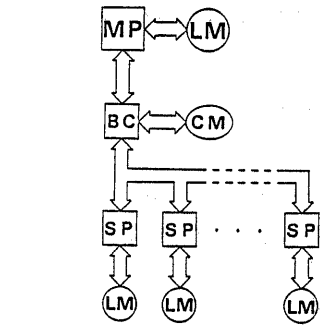
プロセッサ間の結合形態として単一バスを用いたマスタ・スレーブ方式の共有メモリ型並列処理システムの概念図を図１に示す。図１-(a)はマスタプロセッサ（ＭＰ）とＮ個のスレーブプロセッサ（ＳＰ）および共有メモリ（ＣＭ）とバスの管理を行うバスコントローラ（ＢＣ）が単一バスによって結合された非クラスタ方式の場合である。各プロセッサは、それぞれローカルメモリ（ＬＭ）を持ちプロセッサエレメント（ＰＥ）を構成する。これに対し図１-(b)は複数のＳＰ群がクラスタ単位にまとめられ、クラスタごとにＢＣおよびＣＭを有している。

プログラムは、システムをＳＩＭＤ（Single Instruction stream / Multiple Data stream）型計算機として用いる場合、ＭＰが全ＳＰのＬＭに一括して転送を行い、ＭＩＭＤ（Multiple Instruction stream / Multiple Data stream）型計算機として用いる場合は、各ＳＰのＬＭに個別に転送する。ＭＰは各ＳＰのＬＭへ個別にデータ転送を行い、ＳＰに演算処理を開始させる。ＳＰからＭＰへの処理結果の転送は、ＳＰが演算処理終了後ＣＭに処理結果を書き込み、ＭＰがそれを読み込むことによって行われる。

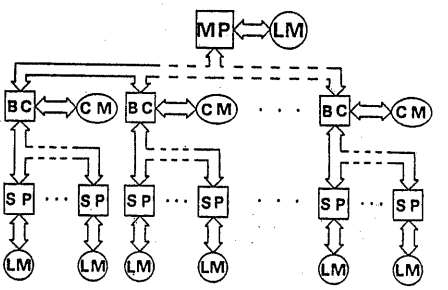
例として、ＳＰ８台のシステムで各ＳＰの処理プログラムは同一で、演算処理時間がデータ転送時間の４倍である場合の基本動作を図２に示す。図中の動作パラメータは次の通りである。

t_p：ＭＰからＳＰへのプログラム転送時間
t_d：ＭＰからＳＰへのデータ転送時間
t_e：ＳＰの処理時間
t_c：ＳＰからＣＭへのデータ転送時間
t_m（$t_{\overline{m}}$）：ＣＭからＭＰへのデータ転送時間

初めに、ＭＰからＳＰへのデータ転送を個別に行った後、ＳＰに演算処理を開始させる。処理が終了しだいＣＭへのデータ転送を行う。非クラスタ方式の場合、No.0のＳＰの処理が終了

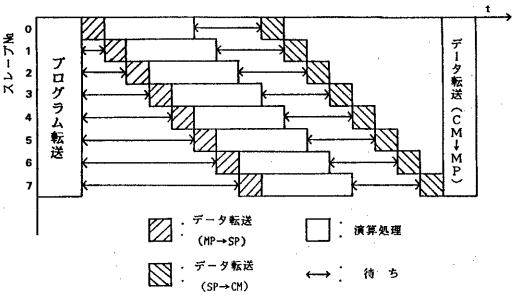


(a) 非クラスタ方式

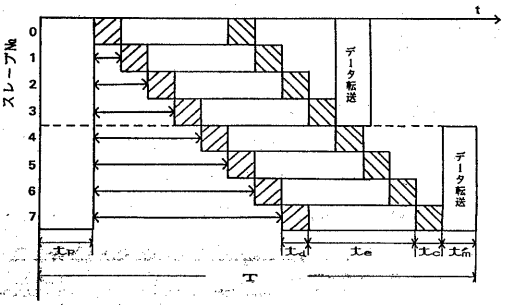


(b) クラスタ方式

図１　共有メモリ型並列処理システム



(a) 非クラスタ方式



(b) クラスタ方式(2クラスタ)

図２　システムの基本動作

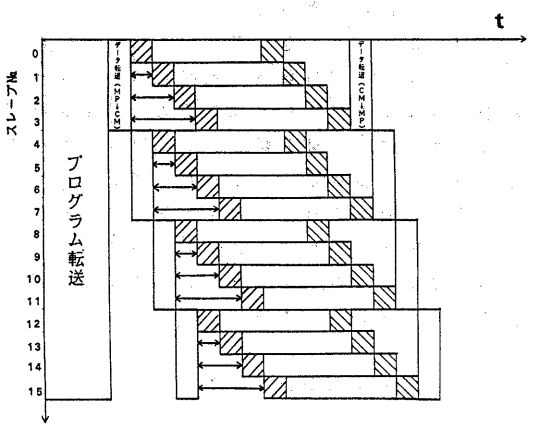


図3　クラスタ方式の基本動作2

してもNo.5からNo.7までのＳＰはＭＰからのデータ転送が終了していないので、バスの使用が認められるまでNo.0のＳＰは待たされる。それに対してクラスタ方式を用いた場合では、データ転送を行うＳＰが、No.5からNo.7までのＳＰとは別のクラスタにあるため、演算処理終了後直ちにＣＭへデータ転送を行うことができる。また、ＭＰがＣＭからデータを読み込む時間もクラスタ単位に分割してできるため、システム全体の処理時間は短縮される。さらに、ＭＰがＳＰへのデータ転送を個別に行うのではなく、一度ＣＭにデータを転送し、各ＳＰがＣＭからデータを読み込むようにすることもできる。ＳＰ１６台を４クラスタに分けたシステムの場合の基本動作を図３に示す。バスの使用効率に関しては、この方式の方が効果的であるが、個別に転送する方式とどちらが良いかは、データの転送量、ＭＰの転送能力、ＢＣの応答時間等によって異なる。

３．プロセッサエレメントの稼働率

3.1 処理時間一定の場合

非クラスタ方式とクラスタ方式の２つの並列処理システムで、各プロセッサの処理時間が一定の場合の処理効率について検討する。

ＭＰが全てのＳＰへのプログラム転送を行い、ＣＭからデータを読み込むまでの時間Ｔは次のようになる。

（１）非クラスタ方式

$$T=\begin{cases} t_p+Nt_d+Nt_c+t_m & (t_e\leqq(N-1)t_d) \\ t_p+Nt_d+t_e+t_c+t_m & (t_e>(N-1)t_d) \end{cases}$$

（２）クラスタ方式

$$T=\begin{cases} t_p+Nt_d+(N/k)t_c+t_m & (t_e\leqq((N/k)-1)t_d) \\ t_p+Nt_d+t_e+t_c+t_m & (t_e>((N/k)-1)t_d) \end{cases}$$

ただし、ＳＰ数をＮ、クラスタ数をｋとする。図４にクラスタ数を変化させた場合のＳＰの稼働率の変化を示す。ここでパラメータＲは処理時間とデータ転送時間の比（t_e/t_d）である。

また、クラスタ数が１の場合は非クラスタ方式（図１-(a)）に一致する。ＳＰが３２台のシステムと６４台のシステムに関して、ＳＰの稼働率Ｐａ（演算処理時間t_e／全処理時間Ｔ）とクラスタ数の関係を図５に示す。各ＳＰでの演算処理は１回とする。演算処理時間がデータ転送時間の100倍（Ｒ=100）のとき、ＳＰが３２台のシステムでは１クラスタ、つまり非クラスタ方式では66.5％、４クラスタで72.6％、８クラスタで73.5％となる。６４台のシステムでは１クラスタで50.5％、４クラスタで57.5％、８クラスタで58.9％となる。Ｒが200のときも、ほぼ同じような傾向を示す。この図から分かるようにＳＰが３２台位のシステムではクラスタ分けしないシステムと比較して４クラスタに分けた場合は１割程度稼働率が向上している。４クラスタに分けた場合と８クラスタに分けた場合を比較するとハードウェアの複雑さが倍以上にな

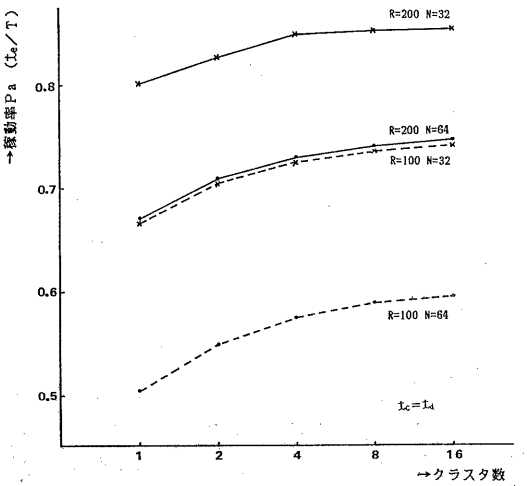


図4　クラスタ数に対する稼動率の変化

るのに比べて稼働率はあまり向上しない。したがってＳＰが３２台から６４台位のシステムでは４クラスタに分けることが適しているといえる。

3.2 処理時間一定の場合の繰り返し処理

ＳＰ３２台を４クラスタに分けたシステムにおいてＳＰの一回の処理時間が一定という条件で繰り返し処理を実行した場合の稼働率を図５に示す。Ｒ＝100で繰り返し処理を行った場合、100回位の繰り返しでは、クラスタ方式を用いると稼働率が２割程度改善されることが分かる。これは、クラスタ方式ではＳＰからＣＭへのデータ転送とＭＰからＳＰのＬＭへのデータ転送がオーバラップして行えるからである。

3.3 処理時間が不定の場合の繰り返し処理

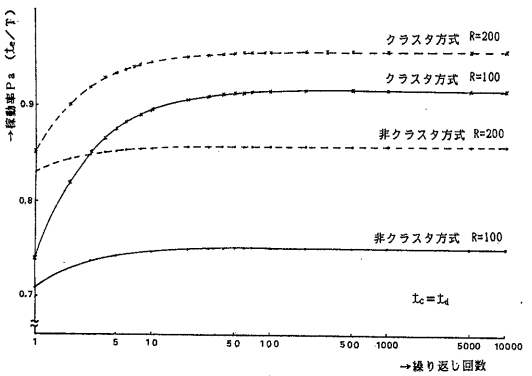


図５　繰り返し処理による稼動率の変化
（処理時間一定）

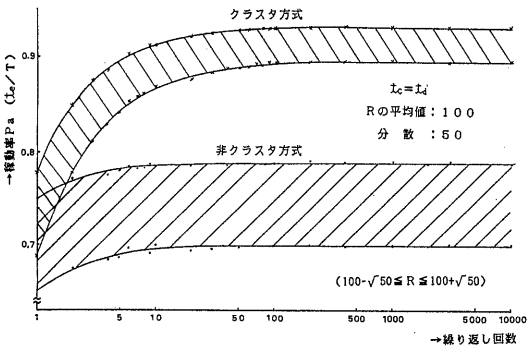


図６　繰り返し処理による稼動率の変化
（処理時間不定）

ＳＰの一回の処理時間が平均100、分散50の正規分布に従い、繰り返し実行した場合のプロセッサエレメントの稼動率を図６に示す。図４の場合と同様にクラスタ方式の場合には稼働率がかなり改善され、変動幅も小さく抑えられることが分かる。変動幅が小さい理由は、非クラスタ方式では最悪の場合、３２台のＳＰが同時に１つのＣＭをアクセスしようとするが、クラスタ方式では１つのＣＭをアクセスするＳＰは最高でも８台だからである。

4．処理効率の測定と検討

4.1 測定用並列計算機

クラスタ方式の動作検討と処理効率の測定を行うためにＭＰと８台のＳＰおよびＣＭを単一バスで結んだ図１-(a)の形式の並列計算機を試作した（写真１）。ハードウェアの仕様を表１に示す。

ＣＭは、プロセッサ間で共有するデータの格納およびプロセッサ間通信用のデータバッファとしてもちいる。ＢＣは、ＣＭをアクセスするためのバス要求の仲裁とＭＰがＳＰのＬＭをアクセスする際のＳＰの管理およびＳＰの処理の終了をＭＰへ通報する役目を担っている。

ＭＰＥ（MASTER PROCESSOR ELEMENT）	台　数	1台
	ＣＰＵ	Ｚ８０Ａ—ＣＰＵ
	ＲＯＭ	８ＫＢ
	ＲＡＭ	５６ＫＢ
	ＰＩＯ	Ｚ８０Ａ—ＰＩＯ（2ポート）
ＢＣ	台　数	1台
	ＣＰＵ	Ｚ８０Ａ—ＣＰＵ
	ＲＯＭ	８ＫＢ
	ＲＡＭ	４ＫＢ
	ＰＩＯ	Ｚ８０Ａ—ＰＩＯ　×４（8ポート）
ＣＭ	ＲＡＭ	３２ＫＢ
ＳＰＥ（SLAVE PROCESSOR ELEMENT）	台　数	8台
	ＣＰＵ	Ｚ８０Ａ—ＣＰＵ
	ＲＯＭ	８ＫＢ
	ＲＡＭ	２０ＫＢ
	ＰＩＯ	Ｚ８０Ａ—ＰＩＯ（2ポート）

表１　測定用システムハードウェア仕様

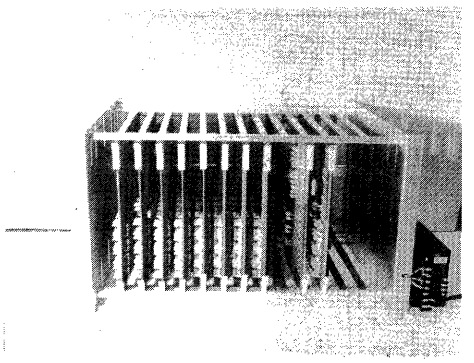


写真1　測定用システム外観

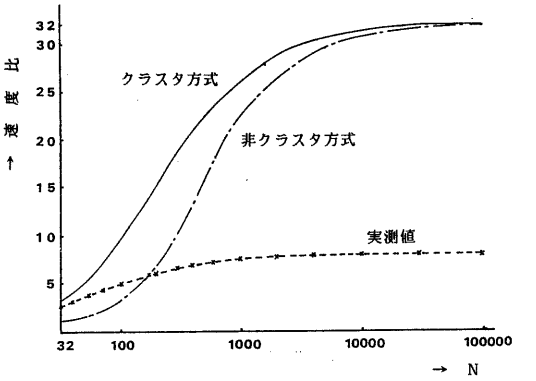


図7　システムの処理速度比

4.2 処理効率の測定

並列処理可能なプログラムの簡単な例として1からnまでの数列の和を求めるプログラムを考える。製作したシステムで実行時間を測定し、1台のプロッセサで実行した場合の処理時間と比較した。プログラムの記述はアセンブラで行った。演算データはすべて4バイトである。図7に示すように処理量が十分に多い場合には8倍のスループットが得られることが分かる。さらに、この測定値を基に32台のSPを4クラスタに分けたシステムとクラスタ分けしないシステムにおける処理効率をシミュレーションで求め比較を行なった。1台あたりの処理量が十分に多い場合は、クラスタ方式と非クラスタ方式の違いはあまりない。しかし処理量があまり多くない場合には、クラスタ方式をとれば非クラスタ方式に比べてかなり処理効率が上がる。たとえば、n＝320の場合、すなわち1台のSPの加算回数が9回の場合、速度比はクラスタ方式で約19、非クラスタ方式で約10となり両者の処理量の違いは2倍近くにもなる。これは、非クラスタ方式ではバスが1本であるのに対してクラスタ方式ではバスが4本あるため共有メモリとのデータ転送にかかるオーバヘッドが緩和されるためである。

5．クラスタ方式の利点と欠点

ここで、バス競合を緩和する方法としてクラスタ方式を用いた場合の利点と欠点をまとめると次のようになる。

利点としては、

(1)一本のバスで結合するプロセッサ数を減らせるため、バス競合が緩和できる

(2)バス仲裁の対象が減りバスアービタの構成と制御が簡単になる

(3)クラスタ単位でのチップ化ができる

(4)システムを階層化できる

(5)クラスタごとに異なるタスクを割り当てることができる

などがある。欠点としては、

(1)異なるクラスタに属するプロセッサ間通信は同一クラスタ内のプロセッサ間通信よりオーバヘッドが大きくなる

(2)ハードウェアが複雑になる

(3)OS設計が複雑になる

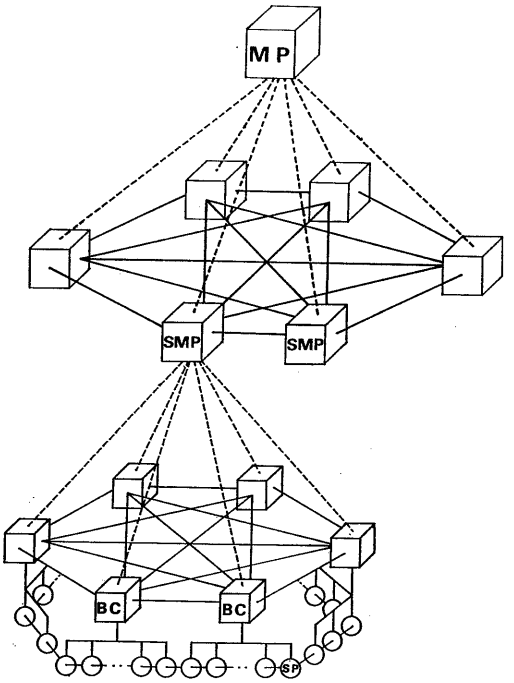
などが挙げられる。

6．クラスタ間完全結合型システム構成

6.1 システム構成

階層化したクラスタ型マルチプロセッサシステムでプロセッサが実用規模のシステムを構成するためにはクラスタ間の接続方法が問題となる。クラスタ間をバスにより完全結合にすれば異なるクラスタに属するプロセッサ間通信が容易に行なうことができ、クラスタ間通信のオーバヘッドを小さくすることができる。完全結合であるために多重通信も可能である。この階層化されたクラスタ間完全結合型システムをMU

GEN(Multiprocessing System with Perfect Connection Network between Clusters)と呼ぶことにする。2階層MUGENのシステム構成を図8に示す。

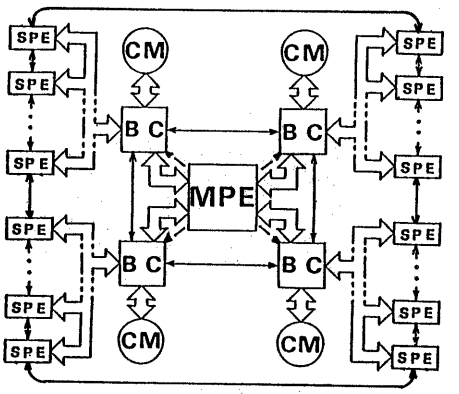


MP: MASTER PROCESSOR
SMP: SEMI-MASTER PROCESSOR
BC: BUS CONTROLLER
SP: SLAVE PROCESSOR

図8　2階層MUGENシステム構成

6.2 システムのハードウェア

現在MUGENの試作機を製作中である。試作システムは、1台のMP，4台のBC，32台のSPおよび4つのモジュールに分けられたCMより構成される（図8）。SP8台・BC1台そして1つのCMモジュールで1つのクラスタを構成する。クラスタ内の全てのSPとCMは、単一バスによって結合され、隣接するSP間はI／Oポート（シリアルポート）で結ぶ。BCはMPやSPからのバス要求の管理とSPの処理状況の監視を行なう。また、すべてのBC間にバスを設け完全結合とし、クラスタ間通信のオーバヘッドを極力小さくする。ハードウェアの仕様を表2に示し、スレーブプロセッサエレメントの外観を写真2に示す。



MPE : MASTER PROCESSOR ELEMENT
SPE : SLAVE PROCESSOR ELEMENT
BC : BUS CONTROLLER
CM : COMMON MEMORY
⟺ DATA・ADDRESS BUS
⟷ I/O PORT LINE

図9　試作システム構成

すべてのプロセッサ間通信はBCにバス要求を出してから行ない、次の3種類の通信がある。

(1)MP・SP間通信

MPからSPへのデータ転送は、MPがSPのLMに直接書き込むか、または通信対象のSPが属するクラスタ内のCMを通して行なう。SPからMPへのデータ転送はクラスタ内のCMを通して行なう。

(2)同一クラスタ内のSP間通信

隣接するSP間の通信はI／Oポートを通して行なう。隣接SP間以外の通信はクラスタ内のCMを通して行なう。

(3)異なるクラスタに属するSP間通信

(2)と同様に、隣接SP間通信はI／Oポートを通して行なう。隣接SP間以外の通信は、CMを通して行なう。ただし、SPが異なるクラスタ内のCMをアクセスする場合は、そのSPが属するクラスタのBCが、他方のBCに要求を出しBC間のバスを開き、SPが直接目的のCMをアクセスする。

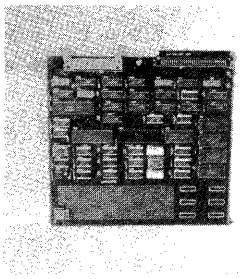


写真2　スレーブプロセッサボード

ＭＰＥ (MASTER PROCESSOR ELEMENT)	台　数 ＣＰＵ ＲＯＭ ＲＡＭ ＰＩＯ ＳＩＯ	1台 Ｚ80Ａ—CPU 8KB 56KB Ｚ80Ａ—PIO　×4 (2ポート) Ｚ80Ａ—SIO (2ポート)
Ｂ　　Ｃ	台　数 ＣＰＵ ＲＯＭ ＲＡＭ ＰＩＯ	1台 Ｚ80Ａ—CPU 8KB 4KB Ｚ80Ａ—PIO　×9 (18ポート)
Ｃ　　Ｍ	数 ＲＡＭ	4モジュール 32KB
ＳＰＥ (SLAVE PROCESSOR ELEMENT)	台　数 ＣＰＵ ＲＯＭ ＲＡＭ ＰＩＯ ＳＩＯ	8台 Ｚ80Ａ—CPU 8KB 20KB Ｚ80Ａ—PIO (2ポート) Ｚ80Ａ—SIO (2ポート)

表2　試作システムハードウェア仕様

6.3 システムのソフトウェア

機会語レベルでプログラムを書くための主な基本マクロ命令を以下に示す。

(1)SLOAD

ＳＰのＬＭにプログラムをロードする。

(2)SRUN

ＳＰにプログラムの実行を開始させる。

(3)EXIT

プログラムの実行を終了してモニタに制御を渡す。

(4)SEND

隣接するＳＰにデータを送る。

(5)RECEIVE

隣接するＳＰからデータを受け取る。

(6)CREAD

ＣＭのデータを読み込む。

(7)CWRITE

ＣＭにデータを書き込む。

以上のマクロ命令は、ＲＯＭの中にサブルーチンの形式で用意されているのでＢＩＯＳコールとして利用できる。

現在、本システムで実行するプログラムの開発するための並列処理記述言語として para-C の構築を行っている[13]。para-Cはｃ言語に、並列処理を行うための基本命令を付加したものである。

7．まとめ

クラスタ方式を用いたバス結合型並列計算機の概要を述べ、処理効率の測定と検討を行った。さらに、クラスタ間完全結合型システムとその試作機のハードウェアおよびソフトウェアについて述べた。試作機は４クラスタからなるＭＵＧＥＮの下位レベルの構成である。この試作機により種々の実験データの採取を行ない、実用規模のＭＵＧＥＮの実現のための解析と検討を行なう予定である。

《謝辞》日頃、御指導戴く東北大学奈良久教授並びに中村維男助教授に深く感謝いたします。

《参考文献》

(1) G. H. Barnes, R. M. Brown, M. Kato, D. J. Kuck, D. J. Slotnick, and R. A. Stockes: "The ILLIAC-IV computer", IEEE Trans. Comput., C-17, 8, pp. 746-757 (1968).

(2) P. M. Flanders, D. J. Hunt, S. F. Reddaway and D. Perkinson: "Efficient high speed computing with the distributed array processor", High Speed Computer and Algorithm Organization, pp. 113-128, Academic Press (1979).

(3) R. W. Hockney, C. R. Jesshope (奥川峻史, 黒住祥祐訳): "並列計算機", 共立出版 (1984).

(4) Anita K. Jones, Edward F. Gehringer: "The Cm* Multiprocessor Project : Research review", Department of Computer Science, Carnegie-Mellom University, CMU-CS-80-131 (1980)

(5) 白河友紀, 影山隆久, 阿部秀彦, 星野力: "並列計算機ＰＡＸ-128", 信学論(Ｄ), J67-D, No. 8, PP. 853-860(昭59-8)

(6) 阿江, 相原: "並列パイプラインＵＮＩＰ" 昭和57年度信学全大, Ｓ１－３

(7) H. T. Kung: "The Structure of Parallel Algorithm", Advances in Computers, vol. 19, pp. 65-112, Academic Press, New York (1980).

(8) S. Horiguchi, Y. Kawazoe and H. Nara: "A Parallel Algorithm for the Integration of Ordinary Differential Equations", Proceeding of the 1984 International Conference on Parallel Pro-cessing, pp. 465-469 (Aug. 1984).

(9) C. A. R. Hoare: "Communicating Sequential Processes", CACM, Vol. 21, No. 8, pp. 666-667 (Aug. 1978).

(10) P. B. Hansen.: "The Programming languag Concurrent Pascal", IEEE Trans. on Software Engineering, SE-1. No. 2, pp. 199-207 (June 1975).

(11) 鈴木, 安部, 堀口, 川添, 重井:"単一バス並列処理システムの効率と試作機の構成" 情報処理学会第28回全大, 2F-2.

(12) 中田, 鈴木, 安倍, 堀口, 川添, 重井 : "クラスタ方式を用いた共有メモリ型並列処理システムの試作", 情報処理学会第30回全大, 5B-3.

(13) 中津山, 中田, 堀口, 重井, 安倍, 川添 : "バス結合型マルチプウロセッサシステムの高位言語", 情報処理学会第31回全大, 3D-5.