計算機アーキテクチャ　68－6
(1987・11・12)

マルチプロセッサシステム MUGENのソフトウェア構成

Software architecture for the Multiprocessor System MUGEN

高木　康志[+]　堀口　進[+]　川添　良幸[++]　重井　芳治[+]
Yasushi Takaki　Susumu Horiguchi　Yoshiyuki Kawazoe　Yoshiharu Shigei

[+]東北大学工学部情報工学科
Department of Information Engineering , Tohoku University

[++]東北大学情報処理教育センター
ECIP , Tohoku University

あらまし　バス結合型のマルチプロセッサシステムでは、バスの競合によりシステムのスループットが低下する。このバス競合を緩和する方法のひとつとして、プロセッサを複数のクラスタに分けるクラスタ方式が提案されている。本稿では、このクラスタ方式を採用した試作マルチプロセッサシステムMUGEN上にインプリメントしたシステムソフトウェアの構成および並列計算機の性能に関して検討している。特に、並列処理を記述できるC言語para－Cおよびそのプリプロセッサ P3Cと、試作システムのオペレーティングシステムの構成について議論する。

Abstract　A clustered multiprocessor system consisted of 32 processing elements has been designed and built. To implement and develop application programs on the multiprocessor system, the system software is significantly important. In this paper, we show the system software implemented on the clustered multiprocessor system (MUGEN). The system software contains a monitor, para-C and a pre-processor para-C (P3C) which translates a serial program into a parallel program written in para-C. The system performance is experimentally evaluated by using the Livermore loops.

1．はじめに

高速な計算機システムを実現する方法として、多数のプロセッサ上で処理を並列に実行する並列計算機の研究が盛んに行なわれている。この並列計算機の実現方法のひとつに、安価で高性能になったマイクロプロセッサを多数用いて、マルチマイクロプロセッサシステムを構成することが考えられ、また実際に様々な構成の試作機が製作されている[1][2][3][4]。

マルチプロセッサシステムにおけるプロセッサの結合方式としては種々のものが提案されているが、広く使用されているもののひとつに共有メモリとバスを用いた方式がある。そのうち、一本のバスに複数のプロセッサを接続する単一バス方式は、ハードウェア量が少なく構成が容易であるが、プロセッサ数が増加するとバス競合によるオーバヘッドが増大し、処理効率が著しく低下する。これを緩和するために、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考案されている[1][2][4]。この方式は、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オーバヘッドが大きくなるが、比較的大規模なシステムに対しても、処理効率の向上に比べてハードウェアの付加量が少なくてすむという利点を持つ[4]。

このような高速処理を目的とした並列計算機では、ハードウェアが複雑になり、種々の資源を有効に利用する必要があるために、従来の計算機以上にシステムソフトウェアの重要性が増している。一般に、処理を高速に実行するためには、並列計算機の結合方法などのハードウェアの特徴に合わせてオペレーティングシステムを構成しなければならない。更に、プログラムを効率良く記述するための並列処理言語も重要である。

本稿では、クラスタ方式バス結合型マルチプロセッサＭＵＧＥＮ[4][5][6]上にインプリメントしたシステムソフトウェアの特徴ならびに並列処理計算機の性能に関して検討している。次章に、試作クラスタ方式バス結合型マルチプロセッサの構成を簡単に示す。第３章では、既にインプリメントされている並列処理言語ｐａｒａ－Ｃ[7][8][9]ならびに従来のＣ言語で記述されたプログラムをｐａｒａ－Ｃに変換するプリプロセッサ（Ｐ３Ｃ）[12]を中心に、プログラミング環境について述べる。また、リバモアループを用いた並列計算機の性能評価を行い、システム効率の評価・検討を行なう。第４章では、試作システムにおいて必要とされるオペレーティングシステムの構成について議論している。

２．クラスタ方式バス結合型マルチプロセッサＭＵＧＥＮ

バス結合によるマルチプロセッサは様々な問題に対して柔軟に適応することができ、汎用性の高い並列計算機を構築できる。バスの結合形態としては、単一バス、クロスバ、複線バスなどが代表的である。それぞれの特徴をまとめると、まず単一バスシステムは、バス調停を行なうロジックが簡潔であり、ネットワークに関するハードウェア量が少なく構成が容易である。また、クロスバは最大でメモリと同数までのプロセッサが同時にメモリアクセス可能である。複線バスはシステム内に複数のバスを有しているので、バスの本数までのメモリアクセスを同時に行なうことが可能である[10]。

バス結合マルチプロセッサでは、プロセッサ数が増加するにつれてバス競合の生じる確率が高くなり、システムのスループットが低下する。相互結合網のコストとマルチプロセッサのスループットの比を結合の価格性能比と定義すると、単一バスは非常によい価格性能比を得ることができ、それに対してクロスバは価格性能比の点からは問題がある。しかし、単一バスは、プロセッサ数が大きくなるに従ってメモリのアクセス競合によりスループット自体が飽和するため、大規模なシステム構築には向かない。これに対してクロスバはメモリアクセスが多くなっても性能が比較的低下せず、スループットが増加する。従って、単一バスとクロスバの中間的な特性を持ち、メモリアクセス競合が小さいシステムを構築することが望まれる。

メモリアクセス競合を緩和するアプローチとして、系内のプロセッサをクラスタ分けする方法がある。このアプローチは価格性能比の点から好ましいと考えられる。ある程度プロセッサ台数が多く競合が生じているとき、価格性能比は単一バスに比較して非常に改善される。しかし、クラスタ分けした場合には、他クラスタの共有メモリにアクセスする場合、いったん１レベル上の階層にあるマスタプロセッサ（ＭＰ）を経由するため、クラスタ内の共有メモリへのアクセスよりも時間がかかる。また、クラスタ外へのアクセスが多いときにはＭＰがネックとなってシステムの性能が低下する。そのため、これを避けるよう注意してプログラムを作成し

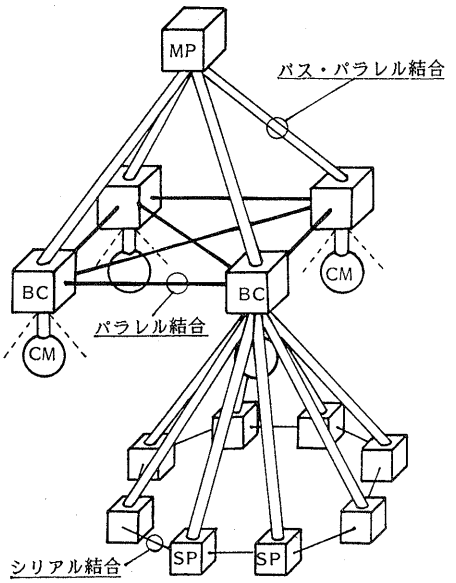

図１　システム構成概念図

MP (MASTER PROCESSOR)	Number	1 Unit
	CPU	Z80A-CPU
	ROM	8 KB
	RAM	56 KB
	PIO	Z80A-PIO×6 (12 Ports)
	SIO	Z80A-SIO (2 Ports)
	CTC	Z80A-CTC
BC (BUS CONTROLLER)	Number	4 Units
	CPU	Z80A-CPU
	ROM	8 KB
	RAM	8 KB
	PIO	Z80A-PIO×8 (16 Ports)
CM (CLUSTER COMMON MEMORY)	Number	4 Units
	RAM	32 KB
SP (SLAVE PROCESSOR)	Number	32 Units
	CPU	Z80A-CPU
	RAM	32 KB
	PIO	Z80A-PIO (2 Ports)
	SIO	Z80A-SIO (2 Ports)
	CTC	Z80A-CTC

表１　試作システムのハードウェア仕様

なければならない。

このクラスタ方式を用いたバス結合マルチマイクロプロセッサの試作システムとして、32台のスレーブプロセッサ（ＳＰ）を持つ試作システムＭＵＧＥＮ[4]を取り上げる。システムの構成概念図を図１に示す。また、試作システムのハードウェア仕様を表１に示す。システム全体を管理するマスタプロセッサ（ＭＰ）の下に４つのクラスタがあり、それぞれのクラスタは、クラスタ内のバスを管理するバスコントローラ（ＢＣ）、プロセッサ間のデータ受け渡しに使用されるクラスタコモンメモリ（ＣＭ）、および並列処理を実行する８つのＳＰで構成されている。

各ＳＰには連続した番号が付けられている。クラスタ０のＳＰ０をスレーブナンバー０とし、以下、クラスタ３のＳＰ７まで各々に０から31までの番号が割り当てられている。これによりＭＰはＳＰ全体を容易に管理できる。

それぞれのＢＣ間はパラレルポートで完全結合され、異なったクラスタ間でのデータ通信を補助している。また隣接するＳＰ間はシリアルポートで接続され、これによってクラスタ内での通信の競合を緩和することができる。バスはアドレスバス16ビット、データバス８ビットから成る。各ＳＰとＢＣ間および各ＢＣとＭＰ間はパラレルポートを通して結ばれる。

ＭＰは現在 RS232Cのシリアル回線(9600bps)によりホストコンピュータと直接に接続され、システムの操作およびプログラムの開発などはホストコンピュータ上で行っている。

各プロセッサの機能と構成について述べる。まず、マスタプロセッサ（ＭＰ）は、プログラムを実行する際に、並列実行が可能なプログラムを32台のＳＰのローカルメモリ(ＬＭ)に転送して演算処理を行わせる。処理された結果は４つのＣＭから読み込む。ＣＰＵにはZ80A-CPUが用いられており、メモリはＲＯＭが８KB、ＲＡＭが56KBで構成されている。メモリの8000H番地からは９バンクに分けられ、ＭＰのＬＭの他、各クラスタのＣＭおよびＳＰのＬＭを直接アクセスすることができる。

スレーブプロセッサ（ＳＰ）は、ＭＰから送られたプログラムを実行してクラスタ内のＣＭに結果のデータを転送する。ＣＰＵにＭＰと同じZ80A-CPUを用いており、ＬＭとして32KBのＲＡＭを持っている。8000H番地以降は自クラスタのＣＭをアクセスする時に用いる。

バスコントローラ（ＢＣ）は、ＭＰがＳＰのＬＭにプログラムをロードする際のＳＰの制御や、ＳＰがＣＭをアクセスする際のバス競合の仲裁などのバスの管理を行う。さらに、ＭＰが起動をかけたクラスタ内のＳＰが処理を終了し、必要なデータが揃ったことをＭＰに通報する処理を行なう。

３．ＭＵＧＥＮのプログラム開発環境

ＭＵＧＥＮ上での応用プログラム開発が効率よく行え、システムの機能を十分に発揮できるためには、高級言語が利用できることが必要不可欠である。ＭＵＧＥＮでは、ホストコンピュータのＣＰ／Ｍ－80上で動作するＣ言語のコンパイラを改良して、本システム上で実行可能なオブジェクトを生成できるコンパイラ（ｐａｒａ－Ｃ）[8]を開発し、利用している。Ｃ言語は、浮動小数点演算などの数値処理が可能で、またアセンブラレベルの細かい記述もできる。アセンブラで記述したプログラムを新しい関数や手続きとしてライブラリに加えることも容易で、さらにライブラリのソースが公開されているため既存の関数を本システム上で実行可能にすることも容易に行える。

実際の応用プログラム開発および実行は、以下のように行なわれる。まず、ホストコンピ

ュータ上でプログラムを作成する。そのプログラムを、ホスト上でコンパイルした後、ＭＵＧＥＮシステム上に転送・実行する。

３．１．並列処理言語ｐａｒａ－Ｃ

ｐａｒａ－Ｃは、並列処理をプログラム上に陽に示す方法を取っており、そのためのキーワードとして cobegin, process, body および forall が追加されている。ｐａｒａ－Ｃのキーワードを表２に、ｐａｒａ－Ｃによるプログラム例を図２に示す。ｐａｒａ－Ｃでは、並列処理の起動は、Ｃのメインプログラムにあたる関数mainによってのみ行なうことができる。cobegin 文と、それに続くprocess 文によって、並列処理が起動される。並列処理の単位はプロセスで、各プロセスはＳＰ上で逐次的に処理される。並列処理される関数は各ＳＰにプロセスとして静的に割り付けられ、process 文によって、対応するＳＰが起動される。body 文はその関数がプロセスの本体であることを示す。プロセスの実行結果として得られるデータは、各クラスタのＣＭを介して、ＭＰ上で実行されるメインプログラムに引き渡される。プロセス間の通信は、関数 send, receive を用いて１対１で行なわれる。

３．２．ｐａｒａ－ＣのプリプロセッサＰ３Ｃ

ｐａｒａ－Ｃでは、並列処理をプログラム上に陽に記述する方法を取る。この方法によって、コンパイラは問題の並列性を自動的に検出する必要がなくなり、また、並列処理の単位をプログラマが任意に決定できるので、実効的なスループットを上げることができる。その反面、並列処理を行なうためには、プログラマが並列処理を直接記述する必要があり、通常のＣ言語で記述されたプログラムをｐａｒａ－Ｃによって並列に実行することはできない。この問題を解決するために、通常のＣ言語で記述されたプログラムをｐａｒａ－Ｃ用のプログラムに変換するプリプロセッサ（Ｐ３Ｃ）[12]を作成した。これを用いることによって、通常のＣ言語で記述されたプログラムを、自動的にｐａｒａ－Ｃ用プログラムに変換し、高速に並列実行することができる。

Ｃ言語で記述された応用プログラムは、図３の手順で処理される。まず、ホストコンピュー

```
main()
{
      int i;
      long n, result, answer, start[32], end[32];

      scanf("%ld", &n);
      for(i = 0; i < 32; ++i)
            start[i] = i*n/32 + 1;
            end[i] = (i+1)*n/32;
      cobegin                          データの転送、ＳＰの起動
            forall(i = 0; i < 32; ++i);
                  process(1; i; sum(start[i], end[i]));
      for(answer = 0, i = 0; i < 32; ++i) {
            hrecieve(i, "%ld", &result);
            answer += result;                    データの受け取り
      }
}
body sum(lower, upper)
long lower, upper;
{
      int i;
      long total;

      for(total = 0, i = lower; i <= upper; ++i)
            total += i;
      hsend("%ld", total);──────────→結果の転送
}
```

図２　ｐａｒａ－Ｃのプログラム例

cobegin	並列処理の起動を宣言
process	関数のＳＰ割り付けおよびプロセスの起動
body	関数がＳＰ上のプロセスであることを宣言
forall	プロセス起動の繰り返し
hsend	ＭＰへデータを送信
hreceive	ＳＰからのデータを受信

表２　ｐａｒａ－Ｃのキーワード

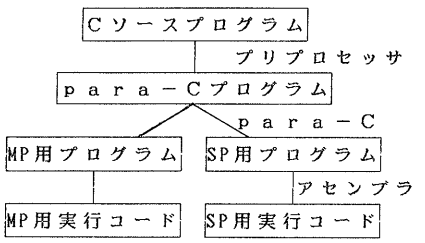

図３　プログラムの処理手順

タ上でプリプロセッサＰ３Ｃによってｐａｒａ－Ｃ用のプログラムに変換され、次に、ｐａｒａ－Ｃコンパイラによって、ＭＵＧＥＮのプロセッサであるZ80のアセンブリ言語に変換される。この時、１つのソースプログラムから、ＭＰ用とＳＰ用の２つのアセンブリプログラムが出力される。このそれぞれをアセンブラによって機械語プログラムに変換し、ＭＵＧＥＮ上に転送して実行する。

ｐａｒａ－Ｃの文法上の制限により、プリプロセッサで処理できるＣ言語のプログラムには、幾つかの制限が設けられる。構造体や共用体は使用できず、浮動小数点演算では、doubleの演算は float(32bits) に変換される。また、プロセスは逐次的に処理され、プロセスからプロセスを呼び出すことは出来ないので、再帰的な関数は、直接並列なプロセスへは変換されない。

３．３．リバモアループによる性能評価

Ｃ言語でプログラムを実行した際のシステムのスループットを測定するために、リバモアループによるベンチマークテストを行なった。

リバモアループ[13]は、約20年前に科学計算コンピュータの FORTRAN 処理能力を測定するためにリバモア研究所で作られたベンチマーク用プログラムである。プログラムは、一般的な大規模計算の中核コードを集めた14個の DO ループで構成される。今日では計算機の処理能力を測定する一般的な検査プログラムとして、広く利用されている。各ループの内容は以下の通りである。

(1) 流体
(2) ＭＬＲ内積
(3) 内積
(4) 帯型連立一次方程式
(5) 三角化消去（下三角形）
(6) 三角化消去（上三角形）
(7) 状態方程式
(8) Ｐ.Ｄ.Ｅ積分
(9) 整数予測
(10) 差分予測
(11) 総和
(12) 差分
(13) ２次元粒子推進
(14) １次元粒子推進

このリバモアループをＰ３Ｃを用いて並列処理検出を行い、ＭＵＧＥＮシステム上での処理時間を測定した。測定結果を表３に示す。表の左側の数値は、ＭＰ１台のみで実行した場合の処理時間、右側の数値は、ＳＰを使用して測定した処理時間である。ここでの処理時間は大量のデータ転送時間も含んでいる。数値が欠けているループは、Ｐ３Ｃを用いてアルゴリズム上並列化が難しいなどの理由で並列処理の実行時間は測定できなかったものである。並列化による速度向上比は、2から17倍と大きなばらつきが見られる。これは、演算のアルゴリズムによって並列化できる度合が違うことと、ＳＰに転送するデータ量が大きく違う事による。

次に、ループ１の場合の速度向上比の台数による変化を図４に示す。台数の増加に従って、速度比の増加の割合が小さくなっていくことが分る。

４．ＭＵＧＥＮ用ＯＳの構成

現在ＭＵＧＥＮシステムは、各プロセッサのメモリ上に常駐するモニタによって制御されている[11]。しかし、マルチプロセッサシステムにおいて実用的なＭＩＭＤのプログラムを動作させるためには、各プロセッサへのプロセスの割付、プロセス間の通信、それに伴うメモリの管理等の、より高度な機能ををシステムで提供しなければならない。また、並列処理プログラムの開発ではデバッグが複雑になるため、プロ

No.	内容	演算時間 (sec.)	
		Z80A	MUGEN
1	流体	973	84
2	M.L.R.内積	944	136
3	内積	1150	140
4	帯型連立一次方程式	606	-
5	三角化消去(下)	1084	-
6	三角化消去(上)	1089	-
7	状態方程式	924	55
8	P.D.E.積分	796	371
9	整数予測	692	81
10	差分予測	348	189
11	総和	409	-
12	差分	535	84
13	2次元粒子推進	713	-
14	1次元粒子推進	994	319

表３　リバモアループの実行時間

グラムの実行時に検査を行なう機能が重要となる。このため、システム上でのプログラムの開発・実行環境を整備する必要がある。試作システムＭＵＧＥＮ用オペレーティングシステム（ＯＳ）は、このような要求に基づいて設計する必要がある。

ＭＵＧＥＮは、クラスタ化によってシステムが階層化されている。またプロセッサが互いに通信を行う際にはシリアルポートなどの複数のパスが存在し、また各パスごとにデータの授受に必要な時間が異なる。これらのハードウェア的特徴を十分に考慮して設計すべきＭＵＧＥＮ用ＯＳに必要な条件をまとめる[14][15]。

(1) システム全体の統一管理（階層内管理、通信パス管理等）
(2) 汎用的なプロセス管理
(3) 整備された開発・実行環境
(4) 簡便な入出力、ファイル管理

ＭＵＧＥＮにおいては、システムの大きさの点で、メモリを多量に消費するような大規模な構成は現実的でない。したがって、ＯＳの基本的な機能に重点を置き、その中でシステムの特徴を生かした構成を考える。具体的には、クラスタ構成に適合したプロセスの管理や、複数のパスを有効に利用するプロセッサ間通信の実行などが挙げられる。これらの条件から検討したＯＳの構成を図５に示す。入出力・ファイル処理は、ターミナルプロセッサ（ＴＰ）を接続し、一括して管理する。

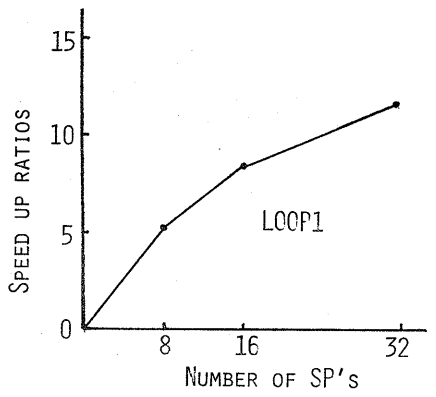

図４　リバモアループの速度向上比

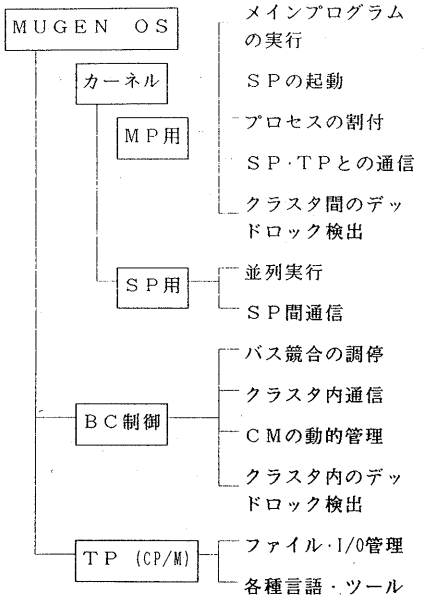

図５　ＭＵＧＥＮ用ＯＳの構成

４．１．ＢＣによるクラスタ管理

システムのクラスタ構成を生かすため、クラスタ内の管理をＢＣによって統括的に行う。内容として、以下のようなことが挙げられる。

1) ＣＭの動的管理
2) ＳＰ間通信およびデッドロック管理
3) バス競合の調停
4) ＭＰ－ＳＰ間の通信管理

まずＣＭの管理については、現在は起動前に静的に各ＳＰの使用する範囲を割当てている。これをＯＳによって一括して動的に管理する。各クラスタ32キロバイトのＣＭを、256個のブロックに分割して管理する。ＭＰおよびＳＰはＣＭを使用する場合、必要なブロック数をＢＣに知らせる。ＢＣは必要な数の連続したブロックが確保できる場合は、プロセッサに先頭ブロックへのポインタを渡し、ＣＭの使用を許可する。このようにＣＭをブロックに分けて管理することにより、管理のテーブルの大きさを抑えることができる。さらに、アクセスのためのポインタも短くなり、その転送による通信時間を減らすことができる。

ＳＰ間の通信方法としては、基本的に各クラスタのＣＭを用いる。しかし、パスの競合によ

る性能低下をできるだけ抑えるため、ポートで接続された各パスを有効に活用しなければならない。各パスのバンド幅が異なるため、通信するプロセッサやデータ量に応じて最適なパスをＯＳが自動的に選択する必要がある。他クラスタのＳＰと通信する場合は、ＭＰが仲介を行なう。また、隣接するＳＰによってパイプラインを構成する場合、他クラスタ内ＳＰも含めて16台以上をリング状に構成することもハードウェア的に可能であるが、この場合も、他クラスタ内ＳＰと接続されるＳＰは例外処理としてＭＰが管理する。

４．２．プロセス割付

ＭＵＧＥＮのＳＰへのプロセス割付について考える。外部の事象によってプロセスが起動される場合に対応して、動的にプロセスをＳＰに割付る機能が必要になる。ＭＵＧＥＮのクラスタ型システムである特徴を生かすためには、相互関連の強いプロセスは同一のクラスタに割付る必要がある。ＳＰの起動はＭＰが直接行なっており、したがってプロセスのＳＰ割付・起動はＭＰが行なう。このため、動的に起動されたプロセスはＭＰが管理し、起動したＳＰに物理的に最も近いＳＰに割当てる。

４．３．プロセッサ間通信

同一クラスタ内のＳＰ間通信について考察する。この場合、ＣＭによる通信のほかに、隣接するＳＰ間のシリアル接続を利用することができる。しかし、表４に示すように、シリアルポートによる通信は転送するバイト数に比例して通信時間が増加する。さらに、隣接していないＳＰとの通信にシリアルポートを用いる場合は、ＳＰ間の間隔に比例した転送時間の増加に加えて、プロセッサの割り込み処理のオーバヘッドも増加する。このため、このパスは数バイトの、極めて短かい情報を転送する場合のみが実用的である。このシリアルポートによるパスは、通信の同期を取るためのメッセージ伝達のみに使う通信方法も考えられる。

	1バイト	10バイト	使用パス
ＳＰ↔ＣＭ（競合なし）	135 μS	165 μS	バス
ＳＰ↔ＳＰ（ＳＩＯ使用）	120	1200	ポート

表４　ＳＰ間の通信時間

５．まとめ

クラスタ方式を用いたバス結合マルチマイクロプロセッサの試作システムＭＵＧＥＮにおけるソフトウェアの構成について、システムソフトウェアを中心に検討した。今後は、ＯＳの全体のインプリメントと、それに伴なうシステム評価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ならびに中村維男助教授に深く感謝いたします。

参考文献

[1] A.K.Jones and E.F.Gehringer: "The Cm* Multiprocessor Project : Research Review", Dep.of Computer Science, Carnegie-Mellon Univ., CMU-CS-80-131 (1980).

[2] 小原："階層構造のＭＩＭＤ型スーパーコンピュータ", 情報処理, vol.25,No.5,pp.480-490,1984.

[3] 白河他："並列計算機ＰＡＸ－128", 信学論(D),J67-D,No.8,pp.853-860 (昭59-8)

[4] 中田他："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", 情処第30回全大, 5B-3.

[5] 中田他："クラスタ方式を用いたメモリ結合型並列処理システムのバス制御方式の検討", 情処第29回全大, 5B-9.

[6] 中田他："クラスタ方式共有メモリ型並列処理システムの処理効率の測定", 昭和60年度信学総合全大, 1712.

[7] 中津山他："バス結合型マルチプロセッサシステムの高位言語", 昭60情処全大 3D-5.

[8] 中津山他："ｐａｒａ－Ｃコンパイラと試作機の性能評価", 昭和61年度電気関係学会東北支部連合大会, 2J-19

[9] 中津山他："並列計算機MUGENにおける並列処理言語", 昭和61年度信学総合全大, 1481.

[10] 中津山他："並列計算機のソフトウェアと性能評価", 信学技報, CAS86-120 (昭61-11).

[11] 高木他："クラスタ式マルチプロセッサのシステムソフトウェア",情処第33回全大,4C-6.

[12] 高木他、"並列処理言語para-Cのプリプロセッサについて",昭和62年度電気関係学会東北支部連合大会
[13] 唐木幸比古："Formula Translator の動向", Computer Today, vol7, No.2, pp35-42, 1984
[14] A.S.Tanenbaum and R.van Renesse : " Distributed Operating Systems ", ACM Computing Surveys, vol.17,No.4,pp.419-470, 1985
[15] G.R.Andrews and F.B.Schneider : " Concepts and Notations for Concurrent Programming", ACM Computing Surveys, vol. 15,No.1,pp.3-43, 1983