HOKKAIDO UNIVERSITY

Title	並列処理によるシミュレータの研究
Author(s)	石動, 善久; 牧野, 圭二; 土肥, 俊
Citation	北海道大學工學部研究報告, 120, 121-130
Issue Date	1984-03-30
Doc URL	http://hdl.handle.net/2115/41867
Type	bulletin (article)
File Information	120_121-130.pdf

Instructions for use

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

北海道大学工学部研究報告
第120号（昭和59年）

Bulletin of the Faculty of Engineering,
Hokkaido University, No. 120 (1984)

並列処理によるシミュレータの研究

石動善久　牧野圭二*　土肥　俊

（昭和58年11月30日受理）

Studies on Simulator Using Parallel Processing

Yoshihisa ISURUGI, Keiji MAKINO and Shun DOI

(Received November 30, 1983)

Abstract

This report presents the architecture of the parallel processor array (PPA) which consists of thirty-four high-performance minicomputers. The PPA is mainly designed for solving systems of differential equations at an effective rate in simulation of continuous systems. Many techniques for parallel processing, such as a shared memory, an address-switching and broadcasting techniques are used to reduce the computation time.

The PPA is applicable to the simulator for solving systems of partial differential equations and discrete systems characterized by stochastic processes. This report also presents its applications to a finite-element simulator and a discrete system simulator paying particular attention to their parallelism.

1. はじめに

連続系のシミュレーションでは，アナログ計算機やシミュレーション言語を用いた汎用ディジタル計算機が広く利用されている。それぞれ，高速性と汎用性に特徴づけられており一長一短がある。

アナログ計算機を用いたシミュレーションの特徴は，つぎに示すようにその実行時にある。

（i） 高速な演算　演算の処理速度が非常に速く，実質的に時定数を1秒～10^{-4}秒まで小さくすることができる。このことは，約1.6kHzまでの現象を実時間でシミュレーションが可能であることを意味する。

（ii） 人間・機械相互の親密性　パラメータの設定，変更，観測出力点の変更，追加，及びシステムモデルの部分修正が容易で随時可能である。これは，シミュレーションの過程が解を見ながらの試行錯誤の過程であることを考えると重要である。

（iii） 実システムとシミュレーションモデルの対応の良さ　モデルは，物理システムを近似して表現しており，またアナログ計算機も各演算要素の機能が近似の上で成り立っている。したがって，定性的には得られた解は信頼できると考えられる。

（ii）及び（iii）は，ディジタル計算機を用いたシミュレーションでもシステム・ソフトウェ

精密工学科　自動制御工学講座
*汎用シミュレータ施設

ア及び積分方式の検討によって満たすことができる。

一方，アナログ計算機を用いたシミュレーションの問題点は，そのままディジタル方式を選択する決定的な要因となるものである。

（ｉ） 演算精度　アナログ要素の静的な単体精度は0.01％程度であり，総合精度では2桁程度悪くなる。しかも，保守・管理の問題に加えてスケーリングの良否が精度に影響を与える。ディジタル方式では，原理的にビット長を大きくして精度を上げることができる。しかし，動的精度については単純に議論できない。

（ii） スケーリング　アナログ要素は，固定小数点演算であり事前に問題の物理的な性質から予測される場合を除いて，大きさに関するスケーリングは非常に難かしい。ハイブリッド型であれば原理的に自動スケーリングは可能であるが，問題の規模によっては設定に多くの時間を必要とし実用的でない。

（iii） 演算の柔軟性　アナログ要素では複雑な非線形演算は一般に困難である。

（iv） 問題の規模の限界　シミュレータとして演算の要素数が固定されるため，その要素数を越える規模の問題は扱えない。

（ｖ） 汎用性の問題　演算が電気回路との相似則によって行なわれるため，抽象化が十分でなく汎用性に乏しい。

ディジタル計算機を用いたシミュレーションでは，（ｉ）～（ｖ）はその長所となるが処理速度に関しては大型計算機を利用した場合でも，問題の規模が大きくなると多くの演算時間を必要として，価格・性能比が悪くなる。

このような，従来のアナログ及びディジタル両方式の欠点を克服する試みとして，1970年代中頃から，並列処理方式による全ディジタル型シミュレータの研究・開発が本格的に行なわれ始めた。

Applied Dynamics 社の AD−10[1,2] は，当初アナログ計算機の非線形演算装置として開発された数種の機能プロセッサからなるシステムである。Kalis 等[3,4] は，マイクロプログラムで定義可能な機能を持った並列処理システム MIMUS を提案した。Cennamo 等[5~7]は，データ・フロー型の機能プロセッサによる高速 DDA システムを開発している。また，慶応義塾大学の KCSS[8,9] もプロセッサの結合方式は異なるがデータ・フロー型並列処理方式のシミュレータである。これらの並列化の方式は，アナログ計算機の演算要素を機能モジュールとして並列化する方式である。

偏微分問題等の非常に大規模な問題を解くことを目的としたものに，WISPC[10]，PACS[11]がある。前者は3次元プロセッサ・アレイであり，後者は2次元アレイ構造を持ち，いずれも大規模な並列処理システムを目指している。

本講座では，故小山昭一先生がマルチプロセッサシステムによるストアード・インターコネクション方式を取入れた DDA 型のシミュレータを提案[11]し，以後並列化の方式，積分方式，上位言語，及びプロセッサ間の結合方式の研究を行ない，汎用マルチプロセッサシステム HARPS（情報工学専攻に設置）上で連続系シミュレータを実現した[13~16]。

一方，1977年以来汎用シミュレータ施設の新しい高速システム・シミュレーション装置 HOSS（Hokkaido university high−speed System Simulator）の開発が三井造船株式会社との共同で進められ，1980年3月に完成している[17~20]。

この報告は，1975年以来，本講座で進めてきた並列処理方式によるシミュレータの研究における並列化の方式とその実現方法について，連続系シミュレーションの高速化を主たる目的として，開発された HOSS の並列演算装置を中心に述べる。また，並列演算装置の応用として，現在進

行中の有限要素シミュレータ及び離散系シミュレータの並列化の方式を述べる。

2. システム構成

この章では，HOSS のハードウェア構成について述べる。

2.1 HOSS の概要[17~20]

HOSS は3層の階層構造を持ち，1台のホスト・プロセッサ，4台の制御プロセッサ，及び34台のプロセッサからなる並列演算装置（PPA：Parallel Processor Array）から構成される分散処理型マルチ計算機システムである。図1は HOSS の構成を示す。

ホスト・プロセッサは，DEC 社のスーパーミニ計算機 VAX 11/780である。その他のプロ

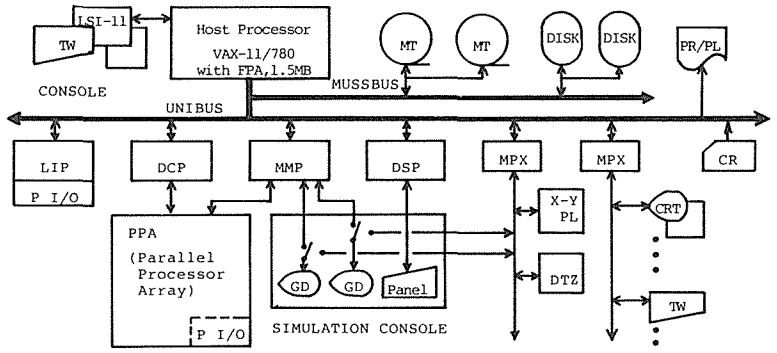


Fig. 1 The configuration of the HOSS system

セッサは，PDP 11/34 相当のアーキテクチャにマイクロプログラムにより目的に応じた特殊命令を付加したミニ計算機である。

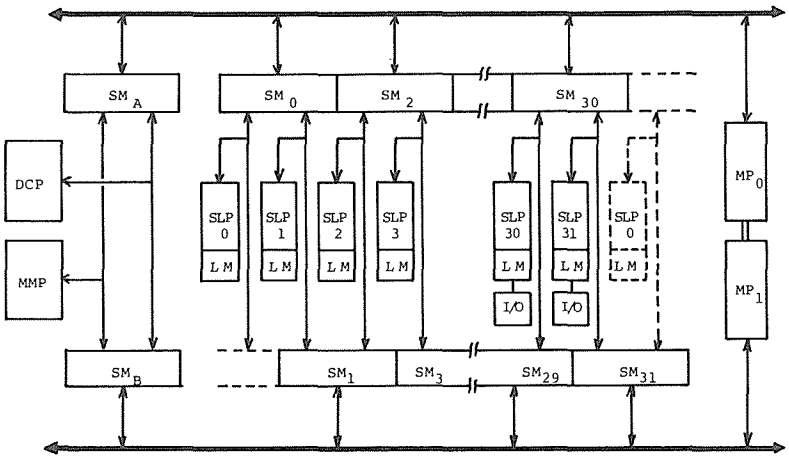
ホスト・プロセッサは，シミュレーション・プログラムの入力・編集，翻訳からシミュレーション実行時の監視及び制御を行なう。

第2レベルの制御プロセッサは，人間−機械系インターフェース・プロセッサとして重要な役割を果す MMP （Man−Machine communication Processor），ホスト・プロセッサと並列演算装置とのデータ転送を制御する DCP（Data Communication Processor），離散系シミュレーションを制御する DSP（Discrete−system Simulation Processor），及び A/D，D/A 変換器，ディジタル入出力ポート等を制御する LIP（Laboratory Interface Processor）である。

第3レベルの並列演算装置 PPA は，主に連立微分方程式で記述されるシステムの高速なシミュレーションを実現するために開発されたものであり，連続系シミュレーションの専用ソフトウェアシステムの下では，アナログ計算機のように動作する。

2.2 PPA の構成

並列演算装置は，2台のマスタ・プロセッサ（MP）と，それらに制御される1次元循環アレイ状に配列した32台のスレーブ・プロセッサ（SLP）から構成される MIMD 型のマルチプロセッサシステムである。これらのプロセッサは，16個のバンクの2つのグループから構成される共有メモリ（SM）により結合されている。さらに，各プロセッサはそれぞれローカル・メモリ



MP : Master Processor SM : Shared Memory DCP : Data Communication Processor
SLP : Slave Processor LM : Local Memory MMP : Man-Machine Communication Processor

Fig. 2 The Configuration of the PPA

（LM）を持っている。図2に PPA の構成を示す。

各 SLP は，32ビット高速浮動小数点演算機構（FFP：内部演算は40ビット，加算は約2.1 μs，乗算は約4.5 μs）をハードウェアで持っている。また，マイクロプログラムによって，数値積分，データ転送，行列演算等の特殊機械命令を備えている。

32台の SLP のうち，4台はA/D，D/A 変換器，及びディジタル入出力ポート等の外部機器とのインターフェースを持ち，オンライン実時間シミュレーションにおける入出力プロセッサとして働く。

(1) 共有メモリ　共有メモリの1つのバンクは，4 KB であり1 KB ずつ4つの領域（FIX_0，FIX_1，EXC_0，EXC_1）から構成されている。このうち FIX_0，FIX_1 は固定アドレス領域で，特に FIX_0 は MP からのブロードキャスト領域である。一方，EXC_0，EXC_1 はアドレス切換えが可能な領域で MP がフラグを制御することによって相互に領域を切換えることができる。

(2) プロセッサ間の結合　プロセッサ間の結合は，共有メモリ方式によっている。共有メモリは，独立な34個のバンクからなり，それぞれが3つの入出力ポートを持ち2台の MP に17個ずつ2グループに分けて接続されている。17個のバンクの SM は，1個は MP，MMP及び DCP が共有しており，他の16個は MP 及び隣り合う2台の SLP が共有している。さらに，隣り合う SLP は異なるグループに属する SM を共有できる形で接続されており，32個のバンクによって32台の SLP は1次元循環アレイ状に結合される。

2台の MP は，それぞれが制御可能な16台の SLP に割込みを起こすことによって SLP を制御する。この割込みは，ブロードキャスト領域の特定のアドレスをアクセスすることにより生じる。逆に，SLP から MP への演算終了等の制御信号は，16台で AND がとられその結果 MP へ割込みがかけられる。2台の MP 間では，その同期は専用信号線を用いて行なわれる。

(3) プロセッサ間のデータ転送　プロセッサ間（MP–DCP，MP–MMP，MP–SLP，及び隣り合う SLP 間）のデータ転送は，34個の SM を用いて行なわれる。MP から SLP へは，さらに，ブロードキャスト転送機能が備わっており，複数の指定された SLP に同一なデータを一時に転送することができる。遠隔の SLP 間の場合，直接的な転送手段がないために MP を

介した方法，あるいは順次 SLP 間の転送を繰返す方法の間接的手段で行なう。

(4) アドレス切換え機能　この機能は，共有メモリの各バンクに独立に存在しているアドレス切換え可能領域を対応する MP がフラグを制御することによって，互いの領域を入れ換える機能である。この機能によって，実際のデータ転送を行なわずに同じバンク内でのデータ転送を行なうのと同じ効果を得ることができる。

3. 連続系のシミュレーションと PPA

3.1 並列化の方式

微分方程式で与えられた数学モデルを表現するのに便利で広く一般に利用されているものに CSMP がある。本シミュレータでは，CSMP 系統のシミュレーション言語 ICOSS[18] によりシミュレーション問題を記述している。したがって，並列化してシミュレーションを実行するにはシミュレーション言語で記述された数学モデルを分割し，各プロセッサに割当てることが必要である。この並列化には，演算要素（加算，乗算等）レベルの分割と方程式（1 次微分方程式）レベルの分割を基本とした方式がある。

演算要素レベルでの分割の場合，さらに割当てに関して，1 つの演算要素を 1 台のプロセッサで処理させる方法[5~9]，及びクリティカルパスを考え負荷均等化による割当てが考えられている。前者は，アナログ計算機と同様に解かれる問題の規模が限定され，また多くのプロセッサ台数を必要とする。後者は，一般に割当て問題を解く必要があること，プロセッサ間のデータ転送の回数が増え同時にプロセッサ間の同期（静的な要素の演算には順序が必要）が複雑になる等の問題がある。

一方，方程式レベルの分割は数値積分における自然な並列性に着目しており，本シミュレータで採用している。この分割は，1 つの積分器とその積分器の入力となる代数演算を単位として分け，つぎに示す特徴がある。

（i） 並列化のための分割が単純である。

（ii） 動的な要素である積分器でモデルの要素間を分けることになるため，時間の制御がほぼ完全にできる。

（iii） 一般にシミュレーションされる物理システムは，積分出力の密な結合からなるサブシステムの集まりであり，それらサブシステムの結合が疎であることが多いため，近接的な接続を重要視した PPA に向いている。

並列化の方式は，ハードウェア構成と密接な関係があり，（iii）についてはむしろ，PPA の構成を決定づけた理由の 1 つと言う方がよいかもしれない。

分割されたブロックは，循環アレイにおける隣接プロセッサ間のデータ結合が密になるようにモデルの規模に応じて 1 個あるいは多数個が SLP に割当てられる。実行形式に翻訳されたプログラムブロックは，各 SLP の LM に格納される。

3.2 シミュレーション実行時の PPA の処理

シミュレーションは，MP の制御下で個々の微分方程式単位に行なわれる積分計算とその結果のデータ交換の 2 つのフェーズからなる時間ステップの繰返しにより進行する。図 3 は，その演算実行サイクルを示す。

(1) 積分計算フェーズ　SLP は，初期値，内部で発生した値，あるいは外部機器からの入力値を初期データとして積分 1 フレーム時間の数値積分を行なう。数値積分法は，オイラー法，

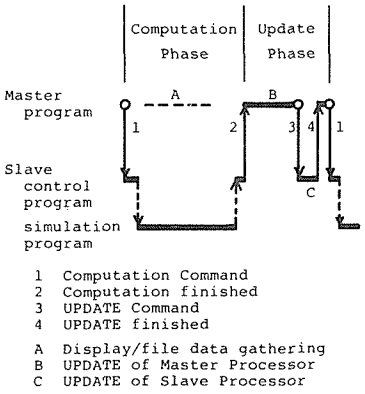


Fig. 3 Execution Cycle of the PPA in Simulation of Continuous Systems

2次のアダムス予測子法，及び3次のルンゲ・クッタ法が選択でき，特殊命令としてマイクロプログラム化されている。この間 MP は，それぞれが管理している16台の SLP の解データを共有メモリから読出し，DCP を介したホスト・プロセッサへの転送，及び解表示のために MMP へ転送する。MP が読出すデータは，アドレス切換え可能な領域内におかれる初期データであり，このフェーズでは変化しない。

(2) データ交換フェーズ　このフェーズでは，積分フェーズの計算結果をつぎのフェーズの初期データとして更新する。初期データの更新には，アドレス切換え機能による更新と MP が介在した遠隔 SLP 間のデータ転送による更新がある。

共有メモリの32個のバンクは，それぞれが2台の SLP に接続されており，また SLP は2個のバンクをアクセスできる。したがって，計算結果を2つのバンクに書込むことによって，その結果は常に両隣りの SLP が参照できる。この同時書込みは，積分命令のマイクロプログラムで処理している。一方，遠隔 SLP 間のデータ交換は，2台の MP が共有メモリから必要なデータを読出し，ブロードキャスト転送によって必要としている SLP が参照可能なバンクに書込むことにより行なわれる。

アドレス切換え可能な領域（EXC_0，EXC_1）の一方を初期データの格納領域とし，他方を積分出力変数の結果領域とすると，フラグの制御によりその論理アドレスが入れ換わる。したがって，ある SLP については，その SLP で演算される積分変数と両隣りの積分変数に関するデータの更新が，データ転送を伴なわずに一時に行なわれることになる。

3.3 連続系シミュレータの性能

積分の1フレームに要する時間Tは，次式によって評価できる。

$$T = t_I \times n_I + t_D \times n_D + t_S + t_O$$

ここで，
n_I：各 SLP に割当てられた最大の積分演算数
t_I：1積分演算当りの正規化された計算時間
n_D：データ転送量の多い方のバスにおける転送データ数
t_D：1データ当りの転送時間
t_S：MP 間の同期に要する時間
t_O：MP における処理とオーバヘッド時間

である。実測の結果は，2次のアダムス予測子法では

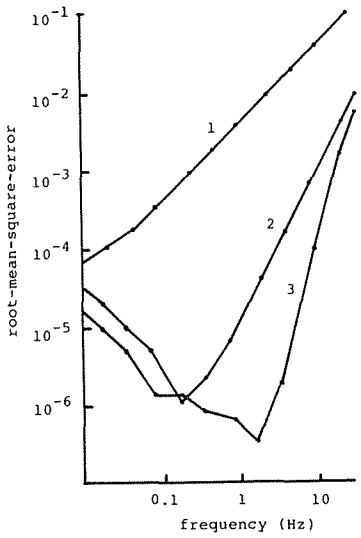
$$T_A = 0.05\ n_I + 0.014\ n_D + 0.04 + 0.3 \quad (ms)$$

また，3次のルンゲ・クッタ法では

$$T_R = 0.148\ n_I + 0.044\ n_D + 0.113 + 0.554 \quad (ms)$$

であった。ここで，t_Iは各積分につき，積分演算及び乗算が各1回であるように正規化されている。この結果によると，MP のオーバヘッド時間t_oが大きく，ファームウェアの改善が今後の課題である。

実時間シミュレータとしての PPA の動特性を実験的に求めたものが図4である。この図は，ここで用いた3つの積分法について周波数に対して自乗平均誤差をプロットしたものである。各 SLP には，積分演算2回，乗算2回が割当てられている。これによって，64次の微分方程式を解く場合，その周波数成分が10 H_Z程度であるシステムの応答が0.1％以下の精度で実時間シミュレーション可能であることが示される。



1 Euler method
2 2-nd order Adams method
3 3-rd order Runge-Kutta method

Fig. 4 Root-mean-square-error characteristics versus frequency of the PPA

4. 応　　用

この章では，PPA の応用として現在研究・開発中の有限要素シミュレータ及び離散系シミュレータについて概説する。

4.1 有限要素シミュレータ[21~23]

偏微分方程式で記述される系のシミュレーション手法として有限要素法がある。有限要素法は複雑な形状の問題にも容易に適用でき，また，解の変化の緩やかな部分では要素形状を大きくすることによって近似の度合をさげることなく節点数を減らすことができる。このため，一様な離散化を行なう差分法と比較して計算時間を大幅に減らすことができる。したがって，有限要素法

を並列処理システム上で実行させることにより，一層の高速化が期待できる。

(1) 並列処理アルゴリズム　有限要素法による偏微分方程式の解法の中心は，離散化により得られる大規模な連立一次方程式の解法にある。それには多くの反復解法があるが，逐次計算で広く用いられている方法に点 SOR 法がある。本シミュレータでは，新たにこの点 SOR 法を並列処理向きに拡張した並列 SOR 法[21]を開発し，PPA を用いて効率よく解いている。並列 SOR 法の収束性については，ある仮定（特定要素がゼロ）のもとで点 SOR 法のそれと一致することが証明されている[22]。この仮定は，有限要素法で導出される連立1次方程式を1次元アレイ状の並列処理システム上で解く場合，ごく自然な形で必ず満たされる。

(2) 分割と割当て　有限要素法では，一旦有限要素モデルが決まるとアレイ状のプロセッサに割当てるのは困難とされた[11]が，基本的には有限要素モデルを PPA の構成と同じ1次元アレイに対応させて分割すれば容易である。すなわち，2次元問題であれば要素の辺に沿って帯状に分割すると，各部は両隣りの情報のみを必要とするだけである。実際には，自動分割を行なうために係数行列の帯状化（節点番号の付け直し）手法を用いている。これには，Gibbs の手法がある。帯行列となった係数行列は

$$\min.\ (m_p,\ N/n_b),\quad m_p: \text{SLP 台数 32},\ N: \text{次元},\ n_b: \text{帯半幅}$$

以下の均等な大きさのブロックに分解（ブロック三重対角行列）し，行ブロックを各 SLP に割当てる。各ブロックの係数行列は，さらにメモリの有効利用のために非零要素のみをベクトル状に圧縮して，対応する SLP の LM 上に格納される。

(3) 各プロセッサの役割　ホスト・プロセッサ：有限要素法を用いて並列処理を行なうための前処理（帯状化，分割，割当て等），及び実行時の解の編集・表示等を行なう。マスタ・プロセッサ：SLP の各処理ルーチンの制御，解析時刻の管理，及び並列 SOR 法の収束の判定を行なう。スレーブ・プロセッサ：割当てられた要素の要素行列の計算，行ブロックの組立てを行ない，割当てられた節点に関して並列 SOR 法の反復計算を行なう。

(4) 性能評価　線形定常問題で節点数1,521点の場合，前処理を除く解析時間を実現されたシミュレータと大型計算機 M-200H で比較すると，PPA：3.34秒，M-200H：1.73秒（点SOR法），M-200H+IAP：1.44秒（消去法）となり，速度では約1/2の性能である。また，本体部分の価格比を1対10と見積っても価格・性能比は5倍近いことがわかる。同一の問題を各 SLP の反復計算を同期をとらずに解いた場合，約25%の向上があった[23]。しかしながら，この非同期並列 SOR 法は，収束性が保証されていないため，今後の課題である。

4.2 離散系シミュレータ

離散系シミュレーションは，時間に対して離散的に事象が発生するシステムを対象としたシミュレーションで，系の表現法の違いにより GPSS[24]と SIMSCRIPT が代表的なシミュレーション言語としてある。GPSS はシステムの中での要素の処理の流れを直接記述でき，待ち行列を含む在庫管理や工程管理等の解析に広く用いられている。試作したシステムは，この GPSS 系統の言語仕様を持つ RISS（Real-time Interactive System Simulator;HOSS の DSP で実行可能）を基礎とした並列処理型離散系シミュレータである。

GPSS のシミュレーションは，システムの中を時間とともに流れる物，すなわちトランザクションに注目し，トランザクションが順次ならんかの変化（処理）を受けてシステムの状態に変化をもたらすことによって進行する。トランザクションに変化を与えるものは，ブロックと呼ばれ，プログラムはトランザクションの生成から消滅までのブロックを表現するステートメントで記述される。

(1) シミュレーション実行時の機能と並列化の方式　シミュレーションの実行時に必要となる機能として，

（i） ネットワーク上のトランザクションの移動などの状態の変化を時間とともに進める機能

（ii） 状態の変化から得られる統計データを編集し，表示する機能

の2つがある。このうち，（ii）に関してはその性質上，並列化は困難である。（i）の処理に関する並列化は，つぎの2つに集約される。

（i） トランザクションについて並列処理を行なう。

（ii） ネットワークを構成するブロックについて並列処理を行なう。

両者を比較した場合，一長一短であるがトランザクションに注目しているGPSS型の言語であること，及び問題の規模に関係するネットワークを分割して処理できることから本シミュレータでは（ii）の方式を採用している。

(2) ネットワークの分割とプロセッサへの割当て　SLP への負荷は，均等になることが望ましく，厳密には最適化問題となるが試作システムではトランザクションの発生から消滅までのクリティカルパスを見つけ，このパスに沿って各プロセッサの処理量が均等になるように，及びトランザクションのプロセッサ間の移動が少くなくなるように考慮した比較的簡単なアルゴリズムを開発し，分割及び割当てを行なっている。

(3) シミュレーション実行時の各プロセッサの役割　ホスト・プロセッサ：下位プロセッサから転送された統計データの処理と表示を行なう。マスタ・プロセッサ：シミュレーション時刻の管理，GATE–GATE OUT 等の対をなすブロックの制御，発生するトランザクションの識別番号の管理，及び統計データの集収を行なう。スレーブ・プロセッサ：MP の管理下でトランザクションの管理と転送，及び実行可能なプログラム・ブロックの処理を行なう。トランザクションの転送は，共有メモリを介して行なう。

(4) 今後の課題　並列化の方式として，GPSS 型のシミュレーションモデルの要素であるブロックに関してネットワークを分割し，並列処理型離散系シミュレータを試作したが，この方式によると，トランザクションの停滞するブロックが，ある SLP に偏より，それが処理速度の低下を招く場合が生じる。この解決法として，ブロックにトランザクションが停滞する時間を考慮した新しい分割及び割当て方法の検討が必要である。

5. おわりに

高速システム・シミュレーション装置 HOSS における並列演算装置 PPA に注目し，そのアーキテクチャ，連続シミュレーションにおける並列化の方式，さらに PPA の応用として偏微分方程式の解析及び離散系シミュレーションにおける並列化の方式について述べた。

ここで述べた並列化の方式は，シミュレーションされるシステムの多くが，密接な接続関係を持ったサブシステムの近接的な結合からなっている点に着目した，ごく自然な分割を基本としている。したがって，1次元循環アレイ状に32台のプロセッサを結合した PPA を用いることによって，比較的単純なアルゴリズムでシミュレーション問題を分割し，各プロセッサに割当てることが可能となった。また同時に，PPA は共有メモリ，アドレス切換え機能，ブロードキャスト機能等により，並列処理で重要な問題であるプロセッサ間のデータ転送及びプロセッサ間の制御を効率よく遂行できるため，連続系シミュレーションのみならず偏微分方程式の解析，離散系のシミュレーションにおける所期の性能を達成している。

ここで述べたシミュレータは，その並列化方式によって大規模なシミュレーション問題に対しても，メモリの制約を除くと処理速度は遅くなるが解くことができる。また一方では，一層大規模なシミュレーション問題を高速に解きたいという要求もあり，この場合には並列処理システムが大規模にならざるを得ない。その時，PPA が採用した構成をそのまま拡張できるか否かは今後の課題である。

本報告は，故小山昭一先生が中心となり，1975年頃より始めた並列処理システムによる全ディジタル型シミュレータの研究に関するものであることを付記しておく。

最後に，本稿を書く機会を与えて下さいました島公脩教授に感謝致します。

参 考 文 献

1) Karplus, W. J.: SIMULATION, (1977), 29, pp. 143-153.
2) Gilbert, E. O., Howe, R. M.: Proceedings of National Computer Conference, (1978), pp. 385-393.
3) Kalis, H., Bantlemann, H., Montag, A.: SIMULATION of SYSTEMS, Proceedings of 8-th AICA Congress, (1976), pp. 277-286.
4) Bantlemann, H., Kalis, H., Montag, A.: Parallel Computers-Parallel Mathematics, (1977), pp. 173-177.
5) Cennamo, F., De Carlini, U.: Microarchitecture of Computer Systems, (1975), pp. 163-170.
6) Cennamo, F., De Carlini, U.: Proceedings of 8-th AICA Congress, (1976), pp. 495-504.
7) Cennamo, F., Sanges, C., Caccaro, R.: Priprint of 9-th IMACS Congress, (1979), pp. 211-217.
8) Yura, E., Yoshikawa, R., Nara, Y., Kimura, T., Aiso, H.: Proceedings of Second USA-JAPAN Computer Conference, (1975), pp. 172-177.
9) Yoshikawa, R. and others: Proceedings of National Computer Conference, (1977), pp. 241-246.
10) Cyre, R., Davis C. J., Frank, A. A.: SIMULATION, (1977), 29, pp. 165-172.
11) 星野：情報処理，20，(昭54)，11，pp. 974-982.
12) Koyama, S, Miura, R.: Priprints of 8-th AICA Congress, (1976), pp. 277-286.
13) 石動，小山：シミュレーション技術研究会，5，(昭52)，5.
14) Koyama, S., Isurugi, Y., Miura, R., Sakurama, M., Sano, T.: EUROMICRO Newsletter, **3** (1977), 4, pp. 75-78.
15) Koyama, S., Miura, R.: Proceedings of IFAC 7-th World Congress, (1978), pp. 781-788.
16) Koyama, S., Sano, T., Miura, R.: 北大工学部研究報告，(昭53)，87.
17) Makino, K., Koyama, S., Miki, N., Iseki, Y., Kobayashi, H., Sakai, Y.: Preprints of 9-th IMACS Congress, (1979), pp. 171-177.
18) Iino, Y., Makino, K., Iseki, Y., Koyama, S., Miki, N.: Preprints of 9-th IMACS Congress, (1979), pp. 261-270.
19) Koyama, S., Makino, K., Miki, N., Iini, Y., Iseki,Y.: Proceedings of IFAC 8-th World Congress, (1981), pp. 1715-1720.
20) 小山，牧野，三木，石動，飯野，井関：シミュレーション，1，(昭56)，1, pp. 42-49.
21) 土肥，小山：電子通信学会論文誌，J 65，(昭57)，4, pp. 464-470.
22) 土肥，小山：情報処理学会論文誌，23，(昭57)，4, pp. 387-395.
23) 土肥：第3回シミュレーション・テクノロジ・コンファレンス，(昭58)，pp. 43-46.
24) 恵羅，岩田，寺西：システムシミュレーション，(昭45)，日刊工業新聞社.
25) 後藤，土肥，石動，牧野：北大工学部研究報告，(昭59)，119.