

図形処理用複合計算機システムの 応用と接続装置について

A COMPUTER COUPLER OF THE PICTURE
PROCESSING ORIENTED POLY PROCESSOR SYSTEM
AND ITS APPLICATIONS

内田 俊一

Shunichi UCHIDA

電子技術総合研究所

Electrotechnical Laboratory

天野 佳之

Yoshiyuki AMANO

慶應義塾大学

Keio University

樋口 哲也

Tetsuya HIGUCHI

1. はじめに

ミニコンピュータやマイクロコンピュータを複数台接続して構成する計算機複合体は、並列処理による処理能力の向上、ファイルや周辺装置などの資源共有、並列処理による処理能力の向上、信頼性の向上、拡張可能性の利点などから注目を集めて来られ、最近のマイクロコンピュータの普及やミニコンの価格低下に伴い、ますます発展し、より高度な応用を目指した複合体がいろいろ試作されている。

これら複合体は、その接続方式の違いから、疎結合のものと密結合のものに大別できる。このうち特に密結合のものは、並列処理による処理速度の向上や、価格性能比（コスト・パフォーマンス）の改善に重点を置いているものが多い。このため、複合体を構成するミニコン等の処理プロセッサ間の関係が密接で、これらの間を接続するバスの構成や接続装置の機能についても、その複合体の用途、すなわち解くべき問題の性質に応じて種々の形態をとるデータ転送・交信を、能率良く行う必要が生じる。

密結合複合体を適用する問題は、問題自身に良質の並列性が内在していなければならぬが、その問題解決のアルゴリズムが複合体の特性に適合していなければ、所期の性能は発揮され得ない。このため、その適用範囲を限定し、問題に共通する性質を考慮して、ハードウェア、ソフトウェアの機能や構成を最適化することが、高い価格性能比のための有効な手段としてよく用いられる。しかしながら、適用範囲を限定することはそれほど容易ではなく、問題と接

続方式の間の関係が既に明確になっているわけでもない。一方、道具としての複合体を見た場合には、なるべく汎用性を持たせておきたいという相反する要求が生じる。このため、一つの解決策として、ダイナミック・マイクロプログラミング等の手法により接続装置に柔軟性を持たせ、これによって相反する要求のそれぞれを、できる限り満たして行こうとする考えがある。

本論文で述べる図形処理用複合計算機システム POPS (Picture Processing Oriented Poly processor System) は図形処理・認識を行うためのシステムである。これの用途は一定限定しているが、具体的に用いられている個々の処理のアルゴリズムには図形処理専用機とするほどの共通した性質は見出せず、上に述べた相反する要求をかがえている。このため、これほど図形処理に共通する性質を考慮しつつ、その接続方式や接続装置についていろいろ実験を行って来られ。その中には 512×512 点、 1024×1024 点、といった大きな二次元配列のデータを扱うために有効な蓄積変換機構等の試作などがあるが、現在、これらに柔軟性を付加し、より広い問題への適用可能性を調べることを目的として、マイクロプロセッサを内蔵した高性能接続装置を試作している。このマイクロプロセッサは書き換え可能記憶へ格納されたマイクロプログラムによって制御されるため、解くべき問題の性質に合わせて、接続装置への特別な機能の追加や変更等が可能であり、必要に応じて処理プロセッサに割当てられるタスク

の一部を分担して実行することもできる。

この接続装置の能力や、POPSの評価を目的として、複数台のミニコンを接続して行列演算やFFTを実行するとし、これらの問題の性質が、POPS全体の制御や接続装置に及ぼす影響について考察し、その効率や速度の改善について試算し、報告する。

〔2.〕 POPSを適用する問題の一般的性質

POPSは、図形処理に関連する種々の問題を対象とするシステムであり、大容量図形データの共有、画像の入力や低レベル処理と認識処理を密接に交信しながら実行する処理の分業、複数のミニコンの接続による処理の並列化などが行えるよう考慮して設計されている。また図形処理に適した複合体の接続方式や、並列処理手法の開発・実験に用いることも目的としている。

ここで扱う、図形処理の問題の特徴は、処理すべきデータ量が莫大であること、演算処理量が多いこと、その処理には多くの場合良復の並列性があることなどが一般的なものとして挙げられる。しかし、個々の手法は、図形処理研究が発展過程にあることから、確立されたものは少なく、絶えず固く改良が加えられているのが現状である。そこで用いられる図形処理アルゴリズムの多くも、現在利用できる演算処理能力の上限を見極め、その限界に近づいたものとなっている。このため潜在的な計算需用はかなり大きく、既存の大型機の処理能力を超えており、図形データの高精度化や色彩情報の利用が進むと、これに応じた演算処理をまかなうには、計算機複合体などのより強力なシステムが必要になると思われる。また、処理の内容も、空間微分や平均化などのほかに、二次元下下T、相関などの信号処理演算、各種の統計量の計算、行列計算、連想処理、リスト処理まで含む幅広いものとなっている。このため、複合体の接続方式や構成の決定にあたり、利用できるような性質を見出すことはなかなか困難である。そこで、ハードウェアについては、以下に挙げるような図形処理に一般的な性質を反映するにとどめ、個々の処理についての特徴は、ファームウェアのようなより柔軟なものの

中で利用する方法が適当である。

本論文では、これらの処理のうちから、その演算処理の性質が比較的明らかで、他の分野でも多用されており、他の実装方式との比較が容易なものを選り出すことにし、その結果二次元下下Tと行列演算をとりあげることにした。これらにより、問題の性質とPOPSの構成、制御方式、および接続装置との関連を調べ、評価する。

二次元下下Tや行列計算、そして多くの図形処理の問題に共通する性質を挙げてみると次のようになる。

- 1) 計算処理に並列性がある。
- 2) 演算処理量が多い。
- 3) データ・サイズが大であり、二次元配列である。
- 4) タスクへの分割が容易であり、一つのタスク中の演算処理量を加減できる。
- 5) 演算処理量を正確に推定することができ、最適化が容易で、また、評価しやすい。
- 6) データが大量であるのに対し、プログラムの手続き部分は小さく、また共有する必要がない。

以上のべた性質は、既に指適されてきたものであり、これらを利用して信号処理や行列計算を目的とする専用機的な並列処理システムの開発が行われた例もあるので、本論文に述べる方式と比較する上でも都合が良い。

1), 2), 4)は、並列処理を効果的に適用するために必要な条件であり、演算処理量が多く、さらに、タスクへの分割が容易であることから、並列処理の制御や、データ転送などのためのオーバーヘッドがうまく吸収でき、処理効率を向上させることが可能となる。また、5)の性質により、予測通りの結果を得やすく、評価の上でも都合が良い。3)の性質は、これらの問題に特有なもので、大容量の高速記憶と大きな蓄地空間の必要性を意味する。また、二次元配列のデータであるため、たいていの場合、データのアクセス蓄地に局所性が少なく、通常のミニコンに採用されているページング等の蓄地変換方式では能率が低下することが多い。接続装置には、これらの性質に適した機能の付加

が必要で、その柔軟性の効果を試す材料となる。大容量記憶は、一般には高価であるから、これを有効に使用するために複合体の共有記憶として、記憶を独立としたものとする方式が有利である。POPSにおいては、この大容量記憶を高精度表示装置のリフレッシュ記憶にも兼用している。

6)は、各プロセス自身にそれぞれ別々にプログラムを持たせればよく、いくつかのプロセッサでプログラムを共有する必要のないことを意味している。プログラムを共有するためには、プログラムを共有記憶の中に格納し、接続装置を介して命令フェッチを行う必要が生じる。このように、少量データを頻繁に転送するべく接続装置を使用すると、高価につくことが多い。POPSは単一バスによる接続方式を採用しているため、このようにバスの使用頻度を上げることは望ましくないので、6)の性質は重要である。

以上、図形処理に一般的と思われる性質と、これを並列処理する場合の利用法の一部、およびPOPSの構成との関係について述べた。

また、具体的処理の一つとして行列計算を挙げたが、行列計算にも種々の問題や解法があり、中には上で述べた特徴に合致しないものもある。このような問題はここでは扱わないこととし、さらに問題を絞って行くことにする。

[3.] システム構成と接続装置

§3.1 POPSの構成

POPSの現在の構成を図1に示す。その構成要素は、PDP 11/45, PDP 11/10, およびNEAC 3200/50の3台の計算機、共有画像記憶(Shared Picture Memory)、これらを接続する共通システムバス(CSB)といくつかの接続装置(カプラー)である。CSBは、PDP 11/10のユニバス(UNIBUS)を、バス拡張装置(BEX)によって拡張したもので、図1に示す67本の信号線から成り、ユニバスと同一の転送シーケンスおよび転送バンド幅を有している。

番地空間は4MBであり、この中に、共有画像記憶、作業記憶(Work Memory)、PDP 11/45の主記憶、NEAC 3200/50の主記憶などが割当てられている。

共有画像記憶は、9ビット×1MBの容量を持つMOSメモリで、画像データの格納に用いる。これのアクセス時間は800nsである。この記憶装置は、3つのポートを持ち、CSBのほか、ビデオ・システムに接続し、高精度表示装置のリフレッシュ記憶としても用いられている。

PDP 11/10の役割は、自分のユニバスを提供し、バス・アービタによりバス使用権の授受を制御するほか、接続装置全体の初期化やCSBに接続している計算機からの割り込の受付を行い、排他処理の制御や、システムの全体的な処理状況の監視を行うことである。その他それ自身で処理の一部を分担することもある。

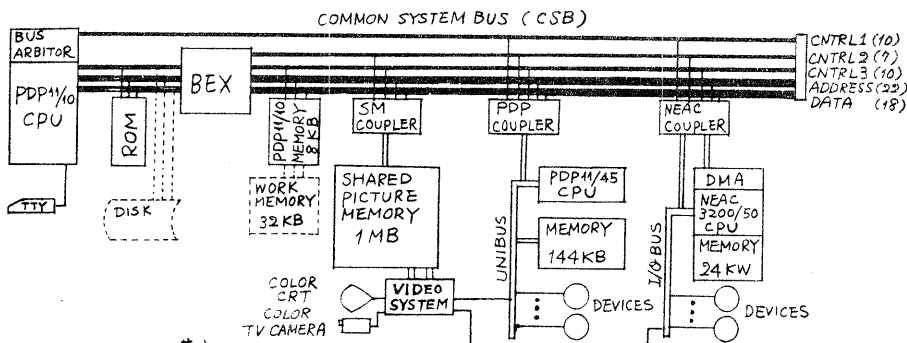


図-1. POPSの構成

BEXは、PDP 11/10のユニバスの番地空間を4MBに拡大するとともに、PDP 11/10がその全番地空間を参照するための番地変換機構を含んでいる。また、大容量の二次元配列のデータをミニコンの小さな論理番地空間から能率良く参照する方法として考案した二次元番地変換機構も含まれており、この方法の効果の実験にも用いられている。

NEAC接続装置 (NEAC COUPLER) は、NEAC 3200/50とCSBを接続するカプラーで、通常のDMAを用いた一次元的なブロック転送機能と、CSB側からCSBの番地空間の一部としてNEAC 3200の主記憶を一語ずつ参照するための機能を持っている。これによってPDP 11/45やPDP 11/10からNEAC 3200の主記憶中のデータを機械語命令のオペランドとして、直接に読み書きすることができる。

PDP接続装置 (PDP COUPLER) は、通常のミニコンと同様のページング方式による番地変換方式を用いて、PDP 11/45 CPUがCSBの番地空間中で直接参照する機能の他に、高速のマイクロプロセッサを内蔵し、ファームウェアによって番地変換、計算機間の文信、データ転送などを行うことができる。これはこのような接続装置の中にマイクロプロセッサを組み込み、接続装置の機能を強化するとともに制御に柔軟性を持たせ、問題の特徴をとらえた処理機能を与えることで、どのような効果が得られるかを実験する目的を持っている。

この接続装置はPDP以外の計算機の接続にも使用でき、単なるデータ転送機能のほかに、高速のデータ処理機能を有することから、高性能接続装置 (Intelligent Coupler) と呼んでいる。

現在のPOPSの構成では2台の計算機が接続されているのみであるが、CSBおよび接続装置は、8台までの計算機もしくは算術プロセッサ等を接続できるように設計されている。

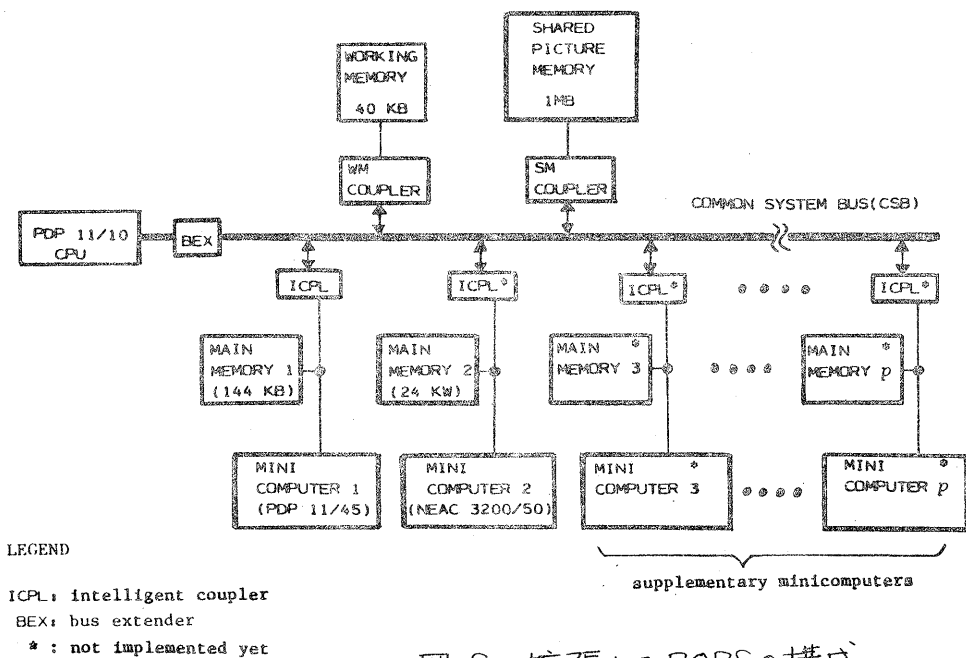


図-2 拡張したPOPSの構成

これらの計算機の接続には、高機能接続装置を用いることができ、これにより、POPS全体の処理能力を増強することができる。

このようにして計算機を追加した場合の構成を図2に示す。この図に示すのは、NEAC 3200/50も高機能接続装置で接続しているほか、同期管理等の作業に用いるための作業記憶を追加している。

§3.2 高機能接続装置の構成

PDPカプラとして使用している高機能接続装置は、CSBとミニコンを接続するためのものである。その構成を図3に示す。内部は図の上半分を占める非同期回路部、下半分のマイクロプロセッサ部の二つに分けられる。

非同期回路部は、ユニバス側(PDP11/45に接続している側)からCSB側をアクセスする機能と、逆に、CSB側からPDP11/45の主記憶をアクセスする機能の二つがあり、いずれの場合にも、ユニバスの番地空間とCSBの番地空間の間での番地変換(アドレス・マッピング)が行われる。ユニバス(UNB)からCSBへのアクセスでは、18ビットのユニバス上の

番地が、通常のミニコンと同様のページングにより22ビットのCSB上の番地へと変換される。この動作は、命令のオペランドフェッチ一回ごとに行われ、これによってPDP11/45は共有画像記憶やNEAC3200の主記憶中のデータを直接参照できる。CSBからユニバスへのアクセスでは、CSB上に、PDP11/45の主記憶が割り付けられている番地パターン(22ビット)が出力されると、カプラが動作し、この番地をユニバス上の番地(18ビット)に変換する。

この時、簡単な記憶保護機構が働き、この番地が、記憶書込上限レジスタ(WTUL REG)と下限レジスタ(WTLL REG)で示される番地の間に入っているかが検査される。そのアクセスが書込み動作であって、かつこの条件を満足していなければ、PDP11/45主記憶への書込み動作は行われず、このアクセスを行った計算機のカプラにはその理由が伝えられる。この他非同期回路部は、PDP11/10や他の計算機に割り込ませかけたり、逆に他からかかって来た割り込ませを受け、PDP11/45へ伝える等の機能を持っている。

非同期的なアクセスによるデータ転送は、PDP11/45から画像記憶へのアクセスを例に

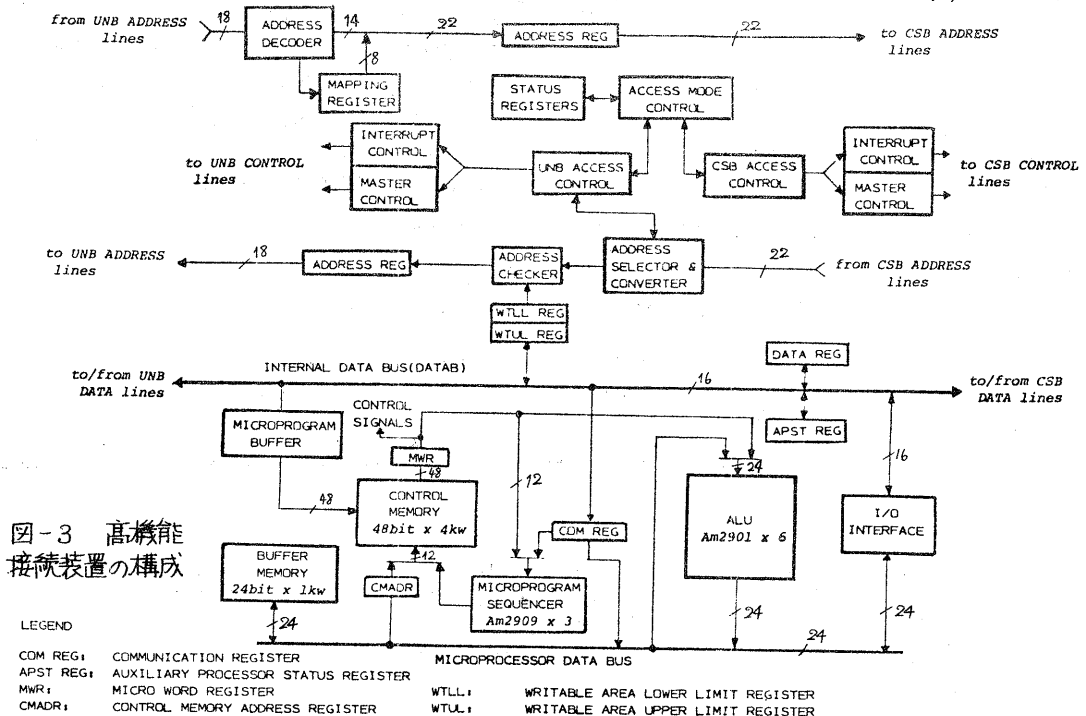


図-3 高機能接続装置の構成

とれば、まずユニバスの使用権をとり、次に番地変換を行い、さらにCSBの使用権をとり、画像記憶をアクセスするという一連の動作が必要なため、一語(16ビット)の転送に、最悪の場合10 μ s以上要する。したがって、このような非同期的なアクセスによるデータ転送は、同期管理情報等の少量のデータを転送するのに適している。しかし、アクセス回数の多い演算処理データの転送には、次に述べるマイクロプロセッサによる転送の方が、高速でCSB上のデータ転送の競合を軽減されるので有利である。すなわち、その二つの転送方法を目的に応じて適宜使いわけることが必要である。

マイクロプロセッサ部は、24ビット幅のALUを中心とするデータ処理部と、12ビットのシーケンサを中心とする命令実行制御部から成る。

データ処理部は、I/Oインタフェースによって、カプラ外部と接続し、非同期回路部に対し転送要求を出すことによって、ユニバス、CSBのいずれの側にもデータ転送ができる。この時、ユニバスおよびCSBの番地レジスタ(ADDRESS REG)にALUの演算結果を書込むことができ、これによって転送データの処理と共に、任意の番地パターンを発生する機能を持つ。また、作業用として24ビット×1K語のバッファ記憶も備えている。

命令実行制御部は、シーケンサのほかに、最大48ビット×4K語まで実装できる制御記憶、制御記憶中のRAM部分へマイクロプログラムを書込むためのマイクロプログラムバッファ、

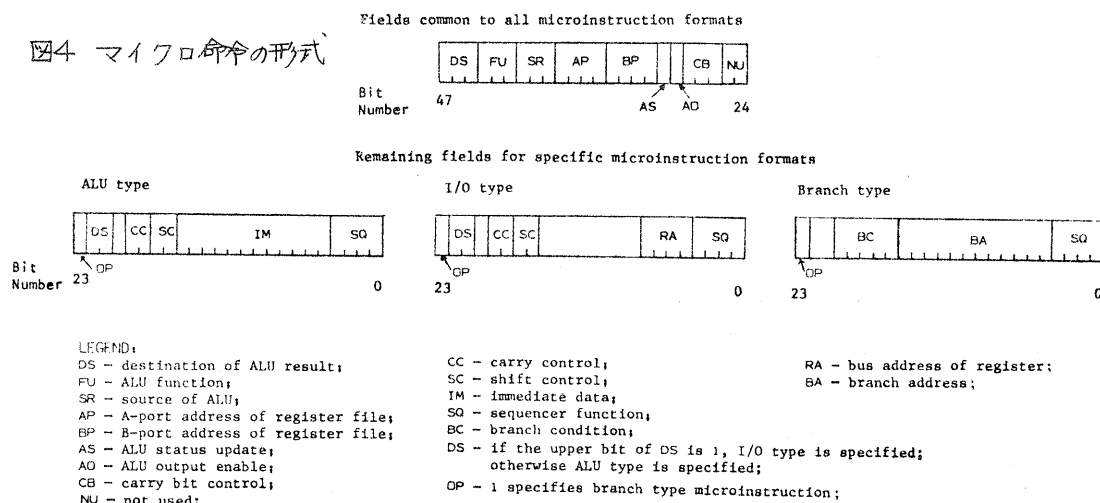
および制御記憶番地レジスタ(CMADR)、POP11/45からマイクロプロセッサへのコマンド(マイクロ命令実行番地)やパラメータを受けとるための受信レジスタ(COMREG)などから成る。

制御記憶は、現在、ROM 256語、RAM 256語、が実装されており、ROM中には、初期化プログラム、マイクロプログラム転送ルーチン、コマンド受付ルーチン等が格納されている。

マイクロ命令は48ビットの幅を持ち、一命令の実行時間は、実装を容易にするために350nsに設定している。命令の形式は、図4に示すように3種類あり、ALU命令、I/O命令、分岐命令から成る。命令を構成する48ビットの、上位22ビットは3種類について共通であり、下位26ビットがそれぞれ異なる。I/O命令は、マイクロプロセッサ部と非同期回路部にある各種レジスタとの間のデータ転送を行う。分岐命令は、ALUの演算結果の状態判定や、非同期回路部の状態判定を行う。

マイクロプロセッサによるカプラ外部とのデータ転送が、非同期回路部を用いるデータ転送と重なることがあるので、非同期回路部の中には、このような競合調整のための回路があり、非同期回路部を用いるデータ転送を優先的に行わせる。データ転送は一語ごとに区切って行われ、待たされた転送は次回に受け付けるようにし、不要なタイムアウト・エラー等を起こさないように配慮している。

図4 マイクロ命令の形式



以上のように、このカプラはそれ自身でデータ処理能力を有しており、その実行速度もミニコンの命令実行速度より1桁程度速くなっている。このため、共有画像記憶やPOP 11/45の主記憶中のデータの並べ換えや、簡単な加減算から成る演算処理等を、ミニコンより高速に実行することが出来る。このため、データ転送時の番地変換などの低レベルの演算のみならず、ミニコンでは計算に手間どるようなFFTのインデクス計算なども容易に実行でき、ミニコンの処理の一部をカプラに分散させることが可能となっている。

また、このカプラは、非同期回路の一部を変更することによって、ユニバス方式を持たないNEAC 3200のようなミニコンに対しても使用できるので、必要に応じて異種計算機を接続することを容易にしている。

§3.3 並列処理の実行方法

並列処理を行う場合のデータやプログラムの配置、データ転送方法、並列処理の制御方式については、このシステムが単一共通バス方式であること、高性能接続装置を有効に活用すること、並列タスクの同期制御が容易であることなどを考慮して決定しなければならない。

POPS では次のような方式を採用し、その実装を検討している。

まず、データやプログラムの配置については次のようにする。

- α1) 共有する図形や行列のデータは、大容量の共有画像記憶に格納する。
- α2) 個々のタスクを実行するプログラムは、それぞれ処理プロセッサの主記憶中に格納する。
- α3) それぞれの処理プロセッサが必要とする共有データの一部は、処理プロセッサの主記憶中の作業領域へ置き、作業領域と共有画像記憶間のデータ転送は高性能接続装置が行う。

このように配置すること、次のような利点が生じる。まずα一に、図形や行列の原データを共有できるばかりでなく一箇所にまとめておくので管理が容易になる。α二に、タスク処理プログラムおよび演算処理を行うデータを

処理プロセッサ中に置くこと、共有画像記憶を直接参照する場合に比べ、処理プロセッサの命令フェッチやデータのアクセスが高速で、さらに共通システムバス(CSB)の使用頻度を最低にすることが出来る。これによって、高価に付くCSBの使用や共有画像記憶のアクセス回数が減り、これらの共有資源の使用の競合が軽減され、システム全体の効率を高め、有効に稼働する処理プロセッサの台数を増加させることが出来る。さらに、共有画像記憶と主記憶中の作業領域との間のデータ転送を高性能接続装置に行わせる際に、マイクロプロセッサのデータ処理能力を活用し、行列計算における転置や、FFTにおける並べ換え、部分行列の抽出し、埋め戻し、などが容易に行える。この結果、論理番地空間の小さなミニコンが大容量データをアクセスする場合に起こる種々の問題を回避出来る。たとえば行列等のデータが、列番号の順に格納されている時、これを行方向に順次アクセスしようとする時、番地空間中の一ページの外へはみ出してしまふことがしばしばある。これを避けようとする時、オーバーヘッドが増大するが、この方法によればこのようなアクセス番地の局所性の無さに起因するような問題を避けることが出来る。α4に、処理プロセッサの動作とマイクロプロセッサの動作は並列実行できる部分があり、これによって一つのタスクを機能分散して処理できるので、処理速度が改善される。

以上述べたように多くの利点があるが、このような利点を実際に発揮されるためには、解くべき問題のアルゴリズムに後述するような並列性に関する条件が必要である。

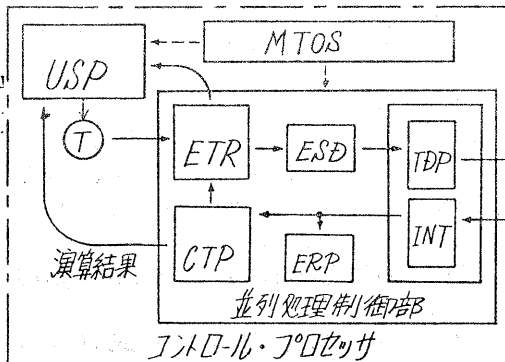
並列処理の実行制御方法については、次のようにする。

- β1) タスクの割付けや同期管理は、一台の計算機を制御プロセッサとして、これにより集中的に管理する。
- β2) ユーザ・プログラムから出される、タスク実行要求の受付と、それらのタスク間の並列性の検出、および処理プロセッサへの割付けは、制御プロセッサが行う。
- β3) タスクの実行に必要な制御情報は、制御プロセッサが処理プロセッサの

主記憶中に書き込み、さらに割込サを掛けて知らせる。

制御プロセッサとしては、機能の高いOSやファイル・システムを既に有するPOP11/45のような計算機を選び、この上でユーザ・プログラムを実行する。ユーザ、プログラム中には同期命令は書かない。そのかわり、ユーザプログラムから逐次的に出されるタスク実行要求を受け付け、そのパラメータ(部分行列名、演算の種類など)を分析して、並列実行可能タスクを動的に検出するための、並列性検出プログラムを準備する。このプログラムはタスク実行要求を先読みし、演算に用いる部分行列名や演算の種類、部分行列中のデータの書き換え(代入)の有無などを分析して、並列性検出グラフを作成する。このグラフに基づき、実行可能タスクを検出し処理プロセッサへ割付け実行させ、実行終了の報告を受けると並列性検出グラフを更新し、新たに実行可能となるタスクを検出する。このようにして動的に実行可能タスクを検出し、その実行順序を再構成して処理プロセッサに割付けて行く。

このような方法を採用することによって、ユーザは、自分のプログラム中に同期命令を書かなくて済み、負担が軽減される。しかしながら、



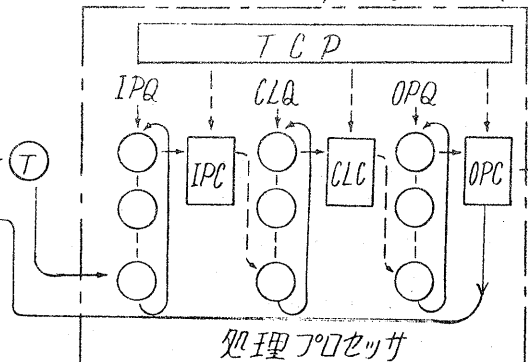
このような検出方法は、かなりの処理時間を必要とするため、タスク処理時間と、並列性検出・割付け時間が接近すると、処理プロセッサの稼働効率を低下させることになる。これを防ぐためにも、解かれるべき問題と解法アルゴリズムの持っている性質をうまく利用することが必要となる。

[4.] popsの適用例と処理効率

前節で述べたような方法を用いて、図2に示したような構成を持った場合の処理の特徴について、行列計算と二次元FFTへの適用例を挙げて示す。まず、行列計算の中で、並列処理に適したブロック処理の可能なものを実行すると仮定し、問題の性質とpops接続、構成方法がどのように関係するかを示す。次に、二次元FFTの演算処理において、高機能接続装置を利用することで、どのような効果が得られるかを、処理速度の改善率で示す。

3.4.1 タスクの性質と処理プロセッサ稼働率との関係

複合体にわたる並列処理においては、演算処理のタスクへの分割方法およびタスク自身の性



USP: ユーザプログラム
SPR: 処理プロセッサ
ESD: 実行順序決定部
INT: 割込処理部
CTP: 終了タスク処理部

MTOS: マルチタスクオペレーティングシステム
ETR: 実行可能タスク検出
TDP: プロセッサ割付け部

TCP: タスク制御部 PPC: 並列処理制御部
IPQ: 入力タスクキュー CLQ: 演算タスクキュー OPQ: 出力タスクキュー
IPC: 高機能接続装置入力管理部 CLC: 演算管理部
OPC: 高機能接続装置出力管理部 T: 処理タスク
ERP: エラー処理部

図-5. タスク実行制御の流れ

度と処理プロセッサの稼働率は密接に関係する。これについて、数百元以上の大型行列を対象とした処理を考え、この関係を示す。大型行列の解法にはいろいろなものがあり、その多くは大型計算機を用いて実行することを前提としているので、アルゴリズムに並列性の少ない場合が多い。ここでは、行列の解法自体よりも、popsの接続方式の評価に重点を置いているので、大型行列をいくつかの部分行列のブロックに分割し、そのブロックを単位として演算を行うような、ブロック処理のアルゴリズムがあるものを対象として選んだ。これらには、連立一次方程式の解法に用いられるガウスの消去法、コレスキーマなどがある。

これらのアルゴリズムでは、一つのブロックを単位として演算を行い、これをくり返して、行列全体の計算を行うことができる。この、ブロックの処理同志の間には並列性が多くあり、ブロック一個の処理を一つのタスクとすると、好都合である。ブロックの大きさ(サイズ)は、ほぼ任意に定めることができるため、一つのタスクの処理時間を変えて、全体の処理効率を上げるような最適化が可能である。

このような性質を備えた問題に的を絞り、pops におけるタスク処理時間を種々の条件について検討する。

まず pops で採用しているような並列性検出および割付方法を用いることによって、次のような制限が生じる。

一つのタスクあたりの並列性検出および割付時間を t_a 、処理プロセッサ台数を p とし、一つのタスクの処理時間を t_c とすれば、常に十分な数のタスクが処理プロセッサに与えられるた

めには

$$t_c > t_a \times p \quad (\text{式1})$$

でなければならない。

pops では、部分行列(ブロック)は、共有画像記憶と処理プロセッサの主記憶との間をカプラーによって転送されるから、一ブロックあたりの転送時間(読み書き双方の和)を t_x とすれば、処理プロセッサが常に処理すべきデータを与えられているためには、

$$t_c > t_x \quad (\text{式2})$$

でなければならない。 t_x は CSB 上の転送時間(共有記憶のアクセス時間を含む) t_{xc} 、ユーバス上の転送時間 t_{xu} およびカプラー内での番地計算その他に要する時間のうちでデータ転送と並列に実行できない部分 t_{mp} よりなる。すなわち

$$t_x = t_{xc} + t_{xu} + t_{mp} \quad (\text{式3})$$

但し、これらは、1ブロック分のデータの読みと書きの双方の時間をまとめたものである。

このうち重要なのは、CSBの専有時間、すなわち t_{xc} であり、タスク処理時間との競合で、CSB上の競合の程度が示される。各処理プロセッサが、この競合による転送の遅れに影響されないためには、プロセッサ台数 p は、次のような値である必要がある。

$$p < t_c / t_{xc} \quad (\text{式4})$$

これらの関係式からわかるように、pops のような構成では、共通システムバス(CSB)の転送速度と、共有画像記憶のアクセス時間が大きな影響力を持っている。

各プロセッサの稼働率を100%近くに上げるためには、タスク処理時間 t_c について、(式1)と(式4)のような条件が必要となる。これら

表-1. タスク処理時間と処理プロセッサ数の関係(概略値)

演算の種類	ブロックサイズ と データ数 k	単位処理 時間	タスク処理 時間 t_c	並列性検 出時間 t_a	CSB専有 時間 t_{xc}	(式-1) k による制限	(式-4) k による制限	処理の 性質
コレスキーマ	32×32 = 1024	$20 \mu s$ データ	655 μs	8 μs	6.5 μs	$p < 81$	$p < 100$	$t_c \propto \frac{k^3}{k^2}$
複素 FFT	1024×1 = 1024	$505 \mu s$ バグワシ	2.6 sec	8 μs	4.3 μs	$p < 325$	$p < 600$	$t_c \propto k \log_2 k$
反復法 ①	32×32 = 1024	$20 \mu s$ データ	21 μs	8 μs	2.3 μs	$p < 2.5$	$p < 8.9$	$t_c \propto k$
反復法 ②	64×64 = 4096	$20 \mu s$ データ	82 μs	8 μs	8.8 μs	$p < 10.3$	$p < 9.3$	$t_c \propto k$

の条件が満たされれば、処理プロセッサを p 台用いれば、処理速度は p 倍となることが期待できる。

これらが実際の問題にフリーでどのような値となるか、popsにおける実際の値を代入してみると次のようになる。各代入値は、いくつかの不確定要素があるため、平均的な値とした。また、簡単のため、データは16ビットの整数とし、算術演算は固定小数点演算とした。^{†)}

まず、カプラから共有画像記憶をアクセスするとき、CSBを専用する時間は、一語あたり $2.1\mu s$ である。1ブロックあたりのデータ数を h 個とし、 $h=1024$ 個(部分行列サイズ 32×32)のとき、各値を、表1に示す。

これを見ると、タスク処理時間 t_c がデータ数に対してどのような関係にあるかによって、有効に移動する処理プロセッサ数が大幅に変わることがわかる。コレスキー法やガウス消去法のように、 t_c が h^2 に比例するものや、FFTのように、 $h \log_2 h$ に比例するものでは、処理の並列性は高く、popsのような単一バス構成でも、理論上は、数十台の処理プロセッサを接続し、有効に移動させ得ることがわかる。しかし、反復法のように、 t_c が h に比例するものでは、(式1)や(式4)による制限のため、反復法②のようにタスクあたりのデータ数を増やしても、共通システムバス上の競合によって、10台程度で限界となる。このような性質の問題に対し、高度な並列処理を行いたい場合は、共通システムバス上の転送時間や共有画像記憶のアクセス時間を短くしたり、バスを複数にするなどの、複合体の基本的な構成の検討が必要となる。しかしながら、表1では、データ1語あたりの処理時間を $20\mu s$ と小さく見積っており、実際には処理プロセッサ台数を、数倍にすることができると考えられる。

†) 行列計算ではこのような仮定は現実的ではないが、並列処理効果を上げる上では、このようにして演算時間を小さく見積もる方が、厳しい条件となる。また、図形処理では、このような条件が一般的である。

§4.2 高性能接続装置による 処理速度の改善

popsでは、大容量の二次元配列データを処理対象とするため、部分行列の切り出しや埋め戻しなどを頻繁に行う。このようなアクセスは、一次元的にみるとアクセス基地に局所性が無く、ミニコン等では一般的に採用されている、ページング方式による参照やDMAを用いたブロック転送などでは効率よく行えない。このために、二次元FFTや行列計算においては、転置などの余分な作業が必要となる。popsでは、高性能接続装置中のマイクロプロセッサにこのような作業を行わせている。また、FFTなどの処理における並べ換えや、FFT実行時の複雑なインテックス計算等も行わせることが出来る。このような高性能接続装置の効果を調べるために、1台のプロセッサによって、二次元FFTを実行した場合の処理時間を求め、通常のブロック転送やチャネルのみの場合と処理速度を比較してみた。また、ミニコンCPUの行う処理の一部を高性能接続装置へ機分散させ、CPUによる処理と並列実行する場合についての改善度を求めた。まず、原データは 1024×1024 英から成る複素データとし、1英は、16ビット $\times 2$ 語から成るものとする。このデータは、共有画像記憶に格納されているものとし、pdp11/45によってFFTを行う。演算は16ビットの固定小数点演算とした。処理は、共有画像記憶から、1列分(1024英)ずつpdp11/45の主記憶へ転送し、そこでバタフライ演算を行い、再び共有画像記憶へ戻すというやり方をとり、演算と転送を並列に行う。このために、主記憶中には、3列分のバッファを設けた。

比較のために演算処理を次のように、5つに分割した。

- 共有画像記憶と主記憶間の1列分のデータ転送
- データの並べ換え(Shuffling)
- 演算すべきデータ対と回転因子のインデックス計算
- バタフライ乗算
- 行列の転置

これらについて、次のような4例の処理を考
える。

- E1) 通常のDMAチャネルを用い、転置
を行う。
- E2) ミニコンはCとDを行い、高性能接
続装置がAとB、および転置に対応
する操作を行う。
- E3) ミニコンは、Dのみを行い、残りは高
性能接続装置が行う。ただし、接
続装置は、ミニコン主記憶のアクセ
スとインデックス計算を並列に実行
しない。
- E4) E3と同じだが、上記のアクセスと
インデックス計算を並列に行う。
(パイプライン処理)

以上の4例についての演算処理時間を、プロ
グラムステップ数から算出し、表2に示した。

E1とE2の差は、転置と並べ換えを高性能
接続装置に行わせている点であり、これによる
処理速度の改善は、1.3倍程度である。E3
は、インデックス計算も接続装置に行わせる場合
で、2.3倍程度に改善される。さらにE4
のように並列化を徹底すると、2.8倍程度まで
改善される。E2の場合の改善率が小さいのは、
FFTに対しては、バタフライ演算とイン
デックス計算に要する時間が長いので、データ転
送や転置の時間の占める割合が小さいことによ
る。データ転送時間が占める割合が、より大
きなものは、E2のような場合の改善率はよ
り大きなものとなると考えられる。

このように処理の一部を分散させることで、
2〜3倍程度処理速度が改善される。

このような方法を、4.1節で述べたタスクの並
列処理と組合せて使用することにより、処理速度の
一層の改善が可能である。FFTのような演
算処理は、4.1節で述べたように、良復の並列
性があるから、この方法によって、全体の処理
速度を $2.8 \times p$ 倍（ p はプロセッサ数）と
することが可能である。

[5.] おわりに

単一共通バスと高性能接続装置によって、大
容量共有記憶と複数のミニコンを接続する構成
は、いろいろな利点を持っている。まずは、
高性能接続装置に、共有記憶とミニコン主記憶
間のデータ転送を行わせることによって、転送
するデータ数を必要最小限に抑えることができ
、この結果、一本しかないバスの能力をフルに
活用できることである。これによって、解く
べき問題の性質に十分な並列性が内在すれば、
一本のバスでも、かなりの数の処理プロセッサ
を接続できる。タスク1個あたりの計算量
データ数に比例する場合には、その比例係数が
問題となるが、処理プロセッサ台数が10台程
度であれば、ほとんどの場合、プロセッサ稼働
率は十分高い値にできる。しかし、このよう
な問題では、問題のタスクへの分割方法や、適
用アルゴリズムに工夫が必要となる。

その他の利点は、高性能接続装置への負荷分散
が出来ることである。これは、内蔵したマイ
クロプロセッサのデータ処理能力の活用である
が、この活用方法は次の二つに分けられる。
その一つは、共有記憶をアクセスするための番
地パターンを操作するものであり、他の一つは、

表-2. FFT実行時間の改善率

例	バタフライ 演算時間 [μs]	1024点の FFT時間 [Sec]	1024×1024 点の2次元 FFT時間 [min]	改善率 [時間比]	処理内容
E1	505	3.5	118	1	1行のDMA転送が30ms。これを、並べかえを 含めると約140ms。行列転置は約720sec.
E2	505	2.6	86	1.37	1行の転送は、35ms。これを並べかえを含め ると約51ms。行列転置は不要。
E3	270	1.5	50	2.35	転送とインデックス計算の双方を接続装置に 行わせる。
E4	219	1.2	41	2.87	転送とインデックス計算をパイプライン処理方式 で行い、並列性を増す。

転送するデータ自身を操作するものである。

これらのうち、蓄地パタンの操作にフリーズは、ミニコンで行うよりも能率が良い。この理由としては、共通バスの蓄地空間は大きく、蓄地パタンのビット幅は、少なくとも22ビット程度であるから、24ビット程度の幅を持たせた接続装置中のALUで演算処理した方が能率がよいこと、一般にミニコンよりマイクロプロセッサの方が実行速度が速く、操作時間を短縮できること、接続装置が付け加えられている位置が共有記憶中をアクセスするのに都合の良いことなどが挙げられる。

この蓄地パタンの操作の応用としては、必要なデータのみに蓄地に関して編集して転送するものがあり、具体的には、前に述べた列番号順に格納された行列の1行を取り出すこと、部分行列の切出し、FFTの時の並べ換えなどのほか、色彩データを各色成分ごとに分けて転送すること、さらに、データがリスト構造を有する場合にポインタをたどりつつ転送することなどが考えられる。

データ自身の操作にフリーズは、本来はミニコンの行うべき仕事であるが、タスク処理時間には転送時間が短ければ、その間に、ミニコンの処理の一部を分担させることができる。その具体的な応用としては、図形データに対しオフセットを加えること、定数倍すること、転送データについて平均値を算出することなどが考えられる。

この他にもいろいろな用途があると思われるが、これが実用になるためには、解くべき問題の性質とこのような処理方式に沿ったアルゴリズムの考案、さらにこれらに基づくタスクへの分割の方法の工夫が重要である。

以上、拡張したpopsにフリーズ述べたが、現在は、図1のような構成であり、各カプラにフリーズ実装を終了したが、まだ調整中である。制御ソフトウェアにフリーズは、その基本的な設計を完了し、並列性検出プログラムの実装を行っている段階である。

最後に、本研究の機会を与えられた電子技術総合研究所 ソフトウェア部 石井治部長、情報システム研究室 棟上昭男室長、ならびに序々、御指導を戴く 慶応義塾大学工学部

相磯秀夫教授に感謝する。また、本論文の作成にあたり、御助言、御指導を戴いた推論機構研究室 眞子ユリ子主任研究官に深謝する。

〔参考文献〕

- 〔1〕 標上世, 「画像処理実験システムと画像処理用多重プロセッサシステム」, 情報処理学会イメージ・プロセッシング研資 6-1, 1976年 5月。
- 〔2〕 野島, 内田, 「図形処理用複合計算機システムののための接続装置と蓄地変換機構」, 情報処理学会計算機アーキテクチャ研資 CA22-1, 1976年 9月。
- 〔3〕 戸川, 「マトリクスの数値計算」, オーム社, 1971年
- 〔4〕 C.V. Ramamoorthy and M.J. Gonzalez, 「A survey of techniques for recognizing parallel processable streams in computer programs」, Proc. of FJCC pp.1~15, 1969
- 〔5〕 M.J. Gonzalez, JR and C.V. Ramamoorthy, 「Parallel Task Execution in Decentralized Systems」, IEEE Trans., vol. C-21, 12, pp.1310-1322, DEC1972.
- 〔6〕 J.O. Eklundh, 「Fast Computer Method for Matrix Transposing」, IEEE Trans. vol. C-21, 7, pp.246-248, JUL.1972.
- 〔7〕 高木, 横井, 「転置行列を高速で得る方法にフリーズ」 信学会, 電子計算機研究会賞, EC73-26 (1973-09), 1973年 9月
- 〔8〕 内田, 「FFTプロセッサ」, 音響学会第15回技術講演会資料, pp.69-87, 1977年 9月