

研究プロジェクト総説

並列処理計算機MUNAPの開発と応用

馬場 敬信⁺

山崎 勝弘⁺⁺

奥田 健三⁺

+ 宇都宮大学工学部

++ 立命館大学理工学部

[概要]

MUNAPは、設計、試作から、応用、評価までを一貫して大学の研究室で行った実験研究用の並列処理計算機である。本計算機では、2レベルマイクロプログラム制御方式による並列処理の制御を基本的な枠組みとしたうえで、種々の非数値処理機能の強化を図っている。本稿では、MUNAPの10年に及ぶ研究経過を概観するとともに、その成果を総括する。

Project Report:

Development of a Microprogrammable Parallel Computer MUNAP

Takanobu Baba⁺, Katsuhiro Yamazaki⁺⁺, Kenzo Okuda⁺

+Faculty of Engineering

Utsunomiya University, Utsunomiya 321
Japan

++Faculty of Science and Engineering,

Ritsumeikan University, Kyoto 603
Japan

Abstract: MUNAP is a microprogrammable parallel computer, designed and developed at a university laboratory. It employs a unique configuration of a two-level microprogrammed, multiprocessor architecture. Several nonnumeric processing units were provided so that the machine can be utilized for a wide spectrum of applications as a universal host computer.

Based on the experiences of hardware development, applications, and architecture evaluation, this paper outlines the MUNAP project, summarizing the major results and emphasizing some lessons learned.

1. はじめに

MUNAPは、設計、試作から、応用、評価までを一貫して大学の研究室で行った実験研究用の並列処理計算機である。

1978年にアーキテクチャの設計を開始して以来、ハードウェアの設計・試作、基本ソフトウェアシステムの設計・試作、さらには言語処理・並列処理等の領域への応用へと発展し、10年余りを経て現在プロジェクト全体としては、ほぼ収束段階にある。

一般に大学の研究環境は、息の長いハードウェアの試作・応用研究には不向きと言われ、実際我々が試作を始めた当時、同種の研究は今のように多くはなかった。このような中で、我々が既存の計算機の活用あるいはソフトウェアシミュレーションという方法を取らず、敢えて本計算機の試作研究に着手した動機を振り返ってみると次のようなことが挙げられる。

第一に、**計算機アーキテクチャの研究を進める上で、既存の商用計算機は、活用しにくいことが挙げられる。**商用計算機は、通常成熟したアーキテクチャ技術の上に開発されるものであり、革新的なアーキテクチャを追求しようとする際に、十分な道具と成り得ないのは当然ではある。

1970年代には、ダイナミックマイクロプログラミングによるアーキテクチャの可変性について活発な研究が行われ、このような研究を可能とする商用のユーザマイクロプログラマブル計算機が現れている。しかし、結局ファームウェアの可変性は、マイクロアーキテクチャによって制限され、既存のマイクロアーキテクチャの制約という掌から飛び出るような研究は難しい。従って、納得のいく研究をするためには、自作するしかないということになる。

第二に、特に並列処理計算機の研究を進める上で、**ソフトウェアのシミュレーションには限界があり、試作による実験・実証的な研究が必要となる**ことが、挙げられる。

シミュレーションでは、並列処理をどこかで逐

次的な処理に置き換えることになり、そこで見落としてしまう課題も多い。実際に、並列プログラムを記述し、実行することによって実感として得られることが多いことを、我々は身を以て感じている。また、MUNAPにおいては、2レベルに分かれたマイクロプログラムと複数のナノプログラムとの間で柔軟な並列処理の制御を行っており、これをソフトウェアで正確にシミュレートするのは困難であるという側面もある。

このようにいろいろな可能性を考えた上で、我々は、並列計算機の設計・試作をして、実証的にその有効性を検証することを決心した。ただし、試作に当たっては、先に述べたような大学における研究環境を考慮し、研究の目的を損なわない範囲でできるだけ実装の難しさを避ける方針を取っている。

以下、本稿では、プロジェクトの概要、システム開発、応用、評価について述べ、最後にプロジェクトの遂行を通じての反省と将来展望に触れる。

主要な成果は既に公表済みであり、ここではプロジェクトにおける開発経験をもとに得た感想などをできるだけ掘り起こして記述するようにしたい。個々の研究成果の詳細については参考文献[1]～[20]を、また、MUNAPプロジェクトの全体像については文献[17]などを参照頂きたい。

2. MUNAPプロジェクト [17]

MUNAPプロジェクトの年表を表1に示す。

プロジェクト全体は、システムの開発と、開発したシステムを用いての応用とに分けられる。

更に、前者は、アーキテクチャの設計、ハードウェアの試作・テストを中心としたハードウェアの開発と、マイクロプログラムの作成支援システムを中心とする基本ソフトウェアの開発とに分けられる。後者の応用については、言語処理、並列処理、その他に大別される。

まず、システム開発が応用に先行しているのは当然であるが、システム開発の中でもアーキテクチャの設計から、試作・調整のように逐次的に進

表 1 MUNAPプロジェクト年表

分類		研究テーマ	実施年度（西暦）											
			78	79	80	81	82	83	84	85	86	87	88	89
システム開発	ハードウェア	<ul style="list-style-type: none"> ○アーキテクチャ設計 ○論理設計・試作・調整 ○MUNAP周辺ネットワーク 												
	ソフトウェア	<ul style="list-style-type: none"> ○マイクロアセンブラ ○モニタ・デバッガ ○ローダ ○エバリュエータ ○ナノプログラム最適化 												
応用	言語処理	<ul style="list-style-type: none"> ○ミニコン機械語 ○MSDL ○L⁶ ○Smalltalk-80 ○Prolog 												
	並列処理	<ul style="list-style-type: none"> ○データベース処理 (ハードディスクを結合) ○数値計算 ○図形処理 (グラフィックディスプレイを結合) 												
	その他	<ul style="list-style-type: none"> ○アーキテクチャ評価 ○マイクロ診断 												

めざるを得ない部分と、これと同時に進められた基本ソフトウェアシステムの開発のように、並行に進められる部分があったことが分かる。進めていく過程で分かったアーキテクチャ上の問題点はこれら並行して走るグループに同時に伝えられ、各々の設計に反映されている。

応用は、基本的に相互独立で、順序関係を持たないが、L⁶の言語処理系の開発とそれを更に発展させた図形処理システムとは逐次的につながっている。また、アーキテクチャの評価もいくつかの応用をこなして初めて可能となった研究である（表中の破線部）。マイクロ診断は、本来は、基本ソフトウェアに含まれる性質のものであるが、本プロジェクトではむしろ知識工学研究の一応用としての性格が強いため、応用に分類してある。

各グループ毎の所要年数には、ばらつきがあるが、3年を要しているものが8件ともっとも多く、「石の上にも3年」の格言を思わせる結果となっている。3年の内訳を大掴みに言えば、検討・システム設計に1年、試作・評価に2年といった所である。また、主な研究では大学院修士課程の学生が核になり、教官の指導のもとに研究を進める体制となっている。

3. システム開発

3. 1 アーキテクチャ設計指針[1][2][3][5]

MUNAPアーキテクチャの設計に当たって、我々が考慮したのはまず試作計算機として、苦勞して実現するに値する魅力的なアーキテクチャを備えていること、そして幅広い応用研究に耐えるだけの柔軟性を持っていることであった。

このような観点から種々考慮した末、「2レベルマイクロプログラム制御方式によるマルチプロセッサ構成」を基本とする並列処理計算機の枠組みを決定した。2レベルマイクロプログラムによる制御の柔軟性と、並列処理による実行性能の向上が基本的な狙いである。また、このような枠組みの元に、それまでの計算機で不得意とされてい

た非数値処理機能を積極的に取り入れることとした。

従って、設計指針は次の3点に要約される。

(1) 並列処理

マイクロプログラムによるきめ細かな制御を前提に、レジスタ転送レベルのMIMD型細粒度 (fine-grain) 並列処理を実現する。

(2) 2レベルマイクロプログラム制御方式

制御記憶をマイクロプログラム記憶とナノプログラム記憶に分け、ナノプログラムは各々の要素プロセッサ(PU)に分散配置する。

(3) 非数値処理指向

機能の直交性・一様性に注意しつつ、非数値処理機能を取り入れる。具体的には、主記憶・PU、PU・PU間を結合するシャフル交換網(SEN)に、ロードキャスト、鏡像変換など種々のデータ交換機能を持たせる。また、主記憶を8バンクにわけ、シャフル交換網とアドレスモディファイアとの組合せにより、バイト単位の変長アクセス、2次元アクセスを可能とする。また、各PUにはビット処理ユニット(BOU)、分割結合ユニット(DCU)などのデータ操作の演算ユニットを装備する。

図1に設計したハードウェアの構成を示す。

3. 2 計算機ハードウェアの設計と試作

ハードウェアの設計・試作に当たっては、研究の本質を損なわない範囲でできるだけ単純化を図った。

集積回路は、実装が容易なLS-TTLを主体に使用し、制御記憶にはNMOSを使用している。通常のALU機能は4ビットスライスチップAm2903でカバーして省力化を図り、一方、非数値処理ユニットにはICをふんだんにつぎ込んで、我々の意図した通りのものを実現している。

2レベルマイクロプログラム方式による制御部の設計は、MUNAPの設計のうちで最も気を使った部分であり、とくにマイクロと複数のナノの順序制御機能相互に関連する部分の設計には、工夫を凝らしている。

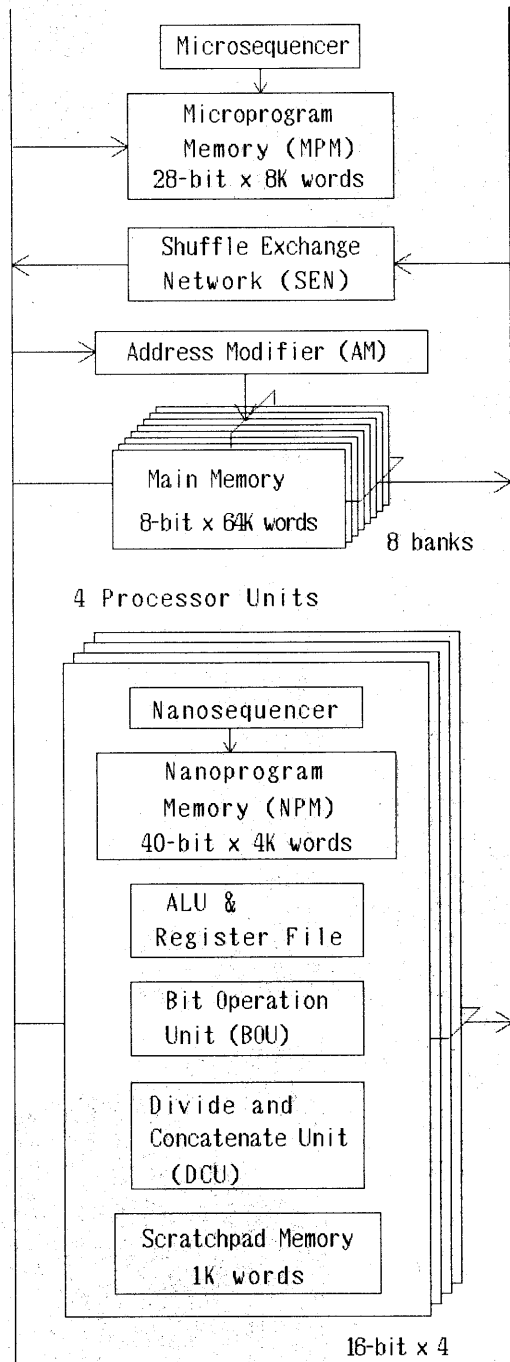


図1 MUNAPハードウェア構成
Fig.1 MUNAP hardware organization

配線については、主記憶など一部でプリント基板を使用しているが、基本的にラッピングにより行っている。マルチプロセッサ部分については、基板を起こすことも考えたが、1台を試作済みであったこと、基板の設計作成に予想以上の時間が取られること、費用の問題などから結局はピンの位置まですっかり同じものをラッピングで複製した。

図2に、MUNAPの概観を示す。その形から、別名、6角堂とも呼ばれるが、6枚の板を大別すると、2枚がマイクロレベルハードウェアを、4枚が4台のプロセッサユニットを実装している。各板は、蝶番で中心部に取り付けてあり、板を開いて自由にチェックができるようにすると共に、中心部での板間フラットケーブルの配線長を抑えるように工夫している。さらに、応用の進展にともないハードディスクおよびグラフィックディスプレイ装置を接続するためのインターフェイスが必要となり、板一枚分の基板が外付けされている。

3. 3 基本システムソフトウェア[4][6][7][9]

MUNAPをマイクロプログラムレベルで活用するための基本システムソフトウェアとして、次のようなものを順次開発した。

マイクロプログラムデバッガは、通常の計算機に於けるコンソール操作を、サービスプロセッサであるECLIPSEより可能とする[3]。

マイクロプログラムアセンブラは、レジスタ転送形式言語により記述された2レベルマイクロプログラムを、翻訳処理する。この際、ナノプログラムの最適化を行う。

作成されるマイクロプログラム量が増大するにつれ、リロケータブルマイクロプログラムローダの必要性が指摘されるようになり、最適化機能を持ったローダが試作された。

種々の応用を通して詳細な動的データを収集し、評価を行うことも、実験研究機として重要である。このために、MUNAPをステップランさせながらデータを収集するエバリュエータが試作された。

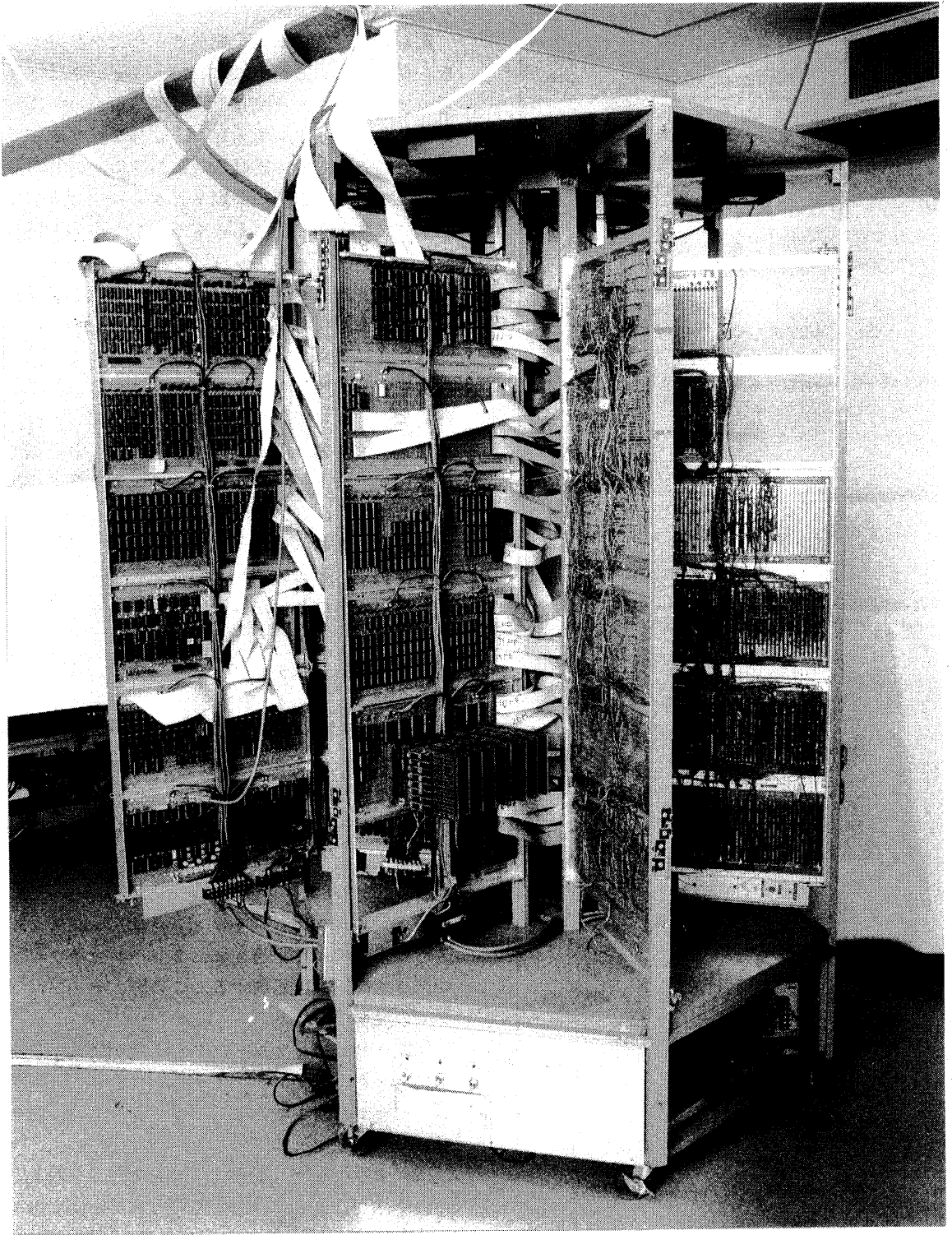


図2 MUNAP概観

Fig. 2 General View of MUNAP

これらの支援ソフトウェアは、図3に示すようにMUNAPに直接結合されたECLIPSEあるいはネットワークを介して接続されたACOS600S上に開発された。

4. MUNAPの応用

MUNAPの応用分野は、大きく言語処理と並列処理に分けられる。前者は、ミニコン機械語、システム記述言語、低レベルリスト処理言語、Smalltalk-80、及びPrologの処理系を含んでいる。後者は、データベース処理、3次元色彩図形処理、数値計算など、大量均質データを対象とした、陽に並列性をもった問題領域に対する応用である。

この他、知識工学からのアプローチとして、MUNAPに対する故障診断システムがある。

以下、各テーマ毎に、その狙い・特徴・実験結果より得られたことなどを要約して述べる。

4. 1 言語処理への応用

4. 1. 1 ミニコンの機械語

本研究では、ミニコンピュータECLIPSE S/130の機械命令セットのエミュレーションにおけるMUNAPのアーキテクチャの有効性の評価を目的とした。

ECLIPSE機械語の命令形式は、語長が16/32/48ビットの3種類で、算術演算命令、論理演算命令、順序制御命令など約100種が存在する。データ型として、固定小数点型(16・32ビット)と浮動小数点型(32・64ビット)がある。本処理系では各機械命令を2レベルのマイクロプログラムで直接解釈実行する。

実験結果から、分割結合ユニットが機械命令のデコードに特に有効であり、実行ステップ数が約40%減少することが分かった[12]。

4. 1. 2 システム記述言語MSDL [8][10]

大規模なシステムプログラムを短期間に、かつ信頼性の高いシステムとして開発するために、MSDL (MUNAP System Description Language) を設計し、その処理系を試作した。

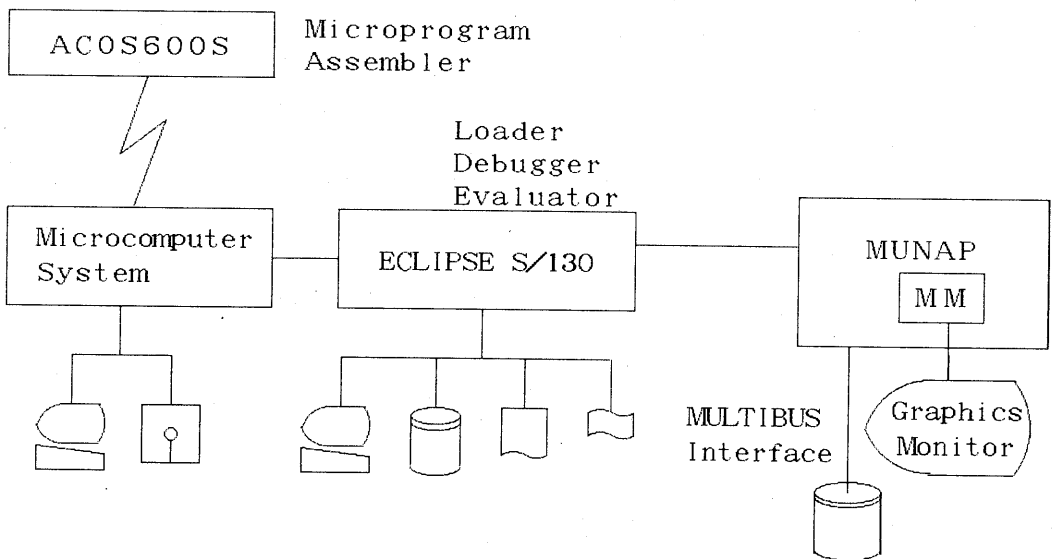


図3 MUNAP周辺の計算機ネットワーク

Fig. 3 Computer network for MUNAP

MSDLはCに準拠した制御構造と演算子を持つ。また、非数値処理用ハードウェアを活用するために、63ビットまでのシフト演算子、交換演算子、及びビット列・文字列処理用の列関数を有している。ファームウェアインタプリタでは、基本マクロ命令によりその小規模化を図ると共に、タグ付きアーキテクチャによりデバッグ支援機能を効率的に実現している。

タグ付きアーキテクチャをタグ無しアーキテクチャと比較評価した結果、制御記憶量は60%に減少し、記憶量の増大や実行速度の低下を招くことなく、効率よくデバッグ支援を行えることが分かった。

4. 1. 3 L⁶を対象としたソフトウェアテストシステム [13]

本研究は、ソフトウェア工学における重要な課題であるソフトウェアテストの問題に対する、アーキテクチャ支援の可能性を明らかにすることを目的として行ったものである。テストの対象となる言語として、比較的簡潔な仕様を持ちながら、柔軟なデータ構造を実現できるL⁶を選んだ。

支援機能としては、データ型に応じたデータ表示、レンジチェック、未定義変数・宙ぶらりんポイント参照検出、実行単位制御、プログラムセグメント単位の実行頻度計測、などを含む。これらの機能は、ファームウェアインタプリタにおいてタグ付きアーキテクチャ、あるいはデータディスクリプタを使用して実現されている。

実験結果より、テスト機能の追加により、86%のエラーが検出され、実行時間の増加は0.56-1.47%に留まることが明らかになった。

4. 1. 4 Smalltalk-80 [16][18]

本研究では低レベル並列処理を活用して、個々のメッセージパッシングの高速処理を図ることを目的とした[16]。通常のバイトコードはスタックマシンを対象とし、スタック操作が全処理の約半分を占めるため、低レベルの並列性の検出が困難である。そこで、メッセージパッシングと一対一

に対応した可変長の間接コードを設定し、その処理において低レベル並列処理を活用した。すなわち、インタプリタで受け手と引数のオブジェクトポイントを並列に決定するためのルーチンを導入した。

実験結果から、オブジェクトポイントのうち69%が並列に決定され、総実行ステップ数がバイトコードに比べて19%減少した。また、バイトコードを単一PUマシンで処理した場合と比較して、処理速度が約2倍であることが分かった。

さらに、メッセージパッシングレベルの高レベルの並列処理を行う計算機をモデル化し、高レベル並列処理と低レベル並列処理の有効性を比較検討した[18]。

4. 1. 5 Prolog [15]

Prologの並列処理については、AND並列、OR並列などが良く知られているが、本研究では、MUNAPのレジスタ転送レベル並列性の活用を目的と絞るため、単一化(unification)の並列処理を行った。

Prologの処理方法としては、Warrenの抽象命令セットを使用するのが一般的であるが、このやり方では、単一化の並列性抽出が難しいため、より高機能の単一化命令を新たに定義した。ファームウェアインタプリタでは、4台のプロセッサで同時に複数の変数の単一化を行う。

実験結果から、平均41%の実行時間の削減の効果が明らかとなっている。

4. 2 並列処理問題領域への応用

4. 2. 1 データベース処理 [11]

本研究では、データ抽象化の概念に基づく関係データベース処理を基本的な目標とした。本方式では、データの定義に沿って、適用可能な操作の集合を生成し、実行時にはこの操作を呼び出すことによって必要な処理を行う。このような形を行うことによって高いデータ独立性が達成される。

データの抽象化に伴う実行時の負荷の増大に対しては、各操作をMUNAP上のファームウェアで実現

することにより、対処する。

実験の結果より、ファームウェア化によって、実行性能を落とすことなくデータの抽象化が実現できることを実証した。

4. 2. 2 3次元色彩図形処理 [19]

本処理系は、先に述べた4.1.3のL⁶処理系を更に拡張する形で実現されている。L⁶を基本にしたのは、図形処理アルゴリズムの中でリスト構造データが活用される場面が多いことによる。

図形処理に応用するため、行列演算・初等関数計算などの機能を追加して拡張した。また、図形表示装置をMUNAPの主記憶を介して外付けした。

実験の結果、移動・回転などの基本的な図形操作に於て、それぞれ汎用ワークステーションの3.0, 7.2倍の実行性能を持つことが明らかになった。

4. 2. 3 数値計算 [17]

数値計算の典型的な課題として、1)大規模疎行列をもつ連立一次方程式、2)高速フーリエ変換、3)連立常微分方程式を取り上げた。1)では非零要素の発生を抑制するための行列の構成、2)ではバタフライ演算の並列実行、ビット逆転関数に鏡像変換の利用、3)では4次のルンゲ・クッタ法により4台のPUを並列使用した。アルゴリズムのもつ並列性をMUNAP上で実現し、これらの計算で平均PU使用台数は、並列処理の効率が落ちる疎行列において、2.0~3.0, その他の場合については3.6~3.9という高い値を示した。

4. 3 その他への応用

4. 3. 1 マイクロ診断 [20]

本研究では、従来のマイクロ診断に知識工学的手法を取り入れて、発生した症状から故障箇所をシステムで自動的に推論することにより、ハードウェアの保守をより系統的に行うことを目的とした。

本システムはマイクロ診断部と故障推定部から

成る。前者はハードウェアユニットをその診断順序に従って階層化し、スタートスモールの考え方に基づいて診断を行い、各ユニット毎に症状を出力する。後者はマイクロ診断で得られた症状と、対象計算機の診断・構造・症状・故障履歴の知識を用いて信号経路に着目した診断を行う。被疑信号集合から正常信号を除去した後、複数ブロックの共通信号に高い優先度を設定して、被疑信号の優先付けを行う。

実験結果から、故障箇所をユニット単位でマクロに限定でき、かつ全被疑信号から約半分の正常信号を除去できること、及び信号経路探索による共通信号は各ユニットに必ず存在し、被疑信号を三段階程度に優先付けできることが分かった。

5. MUNAPのアーキテクチャの評価[12][14]

MUNAPのアーキテクチャを非数値処理指向、多重プロセッサ構成、2レベルマイクロプログラム制御方式の三つの設計方針の観点から、動的評価データを用いて定量的に評価した。対象分野は言語処理からミニコン機械語、MSDL、L⁶を、並列処理問題から数値計算を選んだ。言語処理では、定積分、ハノイの塔、トポロジカルソート、グラフ上の道の探索など5個のプログラムを各言語で記述した。数値計算では高速フーリエ変換とLU分解による連立一次方程式の解法を直接マイクロプログラムで記述した。これらのテストプログラムを実行させ、マイクロプログラムエミュータを用いて種々の動的評価データを収集し、これらを三つの設計方針の観点から分析した。

(1) 非数値処理指向

シャフル交換網では、16ビット単位の巡回シフトとブロードキャストが、PU間転送と直列演算に有効に使用されている。鏡像交換はFFTにおけるMSBとLSBのビット逆転などに使用される。機能制御はシフト数の異なる巡回シフトを統一的に扱う場合に有効である。アドレスモディファイアは1/2/4/8バイトの連続読出しのみが使用された。

分割結合ユニットは、機械命令のデコードにおけるフィールド抽出や高水準言語マシンにおけるタグ操作に特に有効であり、30~40%実行ステップ数が減少する。ビット処理ユニットはプライオリティエンコードを多用する数値計算において有効であり、15%実行ステップ数が減少する。

(2) 多重プロセッサ構成

並列処理の割合を示す平均PU使用台数は、数値計算のようなSIMD型処理で3.6~3.9、言語処理のようなMIMD型処理で2.1~2.5である。これを一般化して考えると、数値計算のように陽に並列性を含む問題ではプロセッサの設置台数に比例した効果を期待できるが、言語処理のように陽には並列性を含まない問題では実行速度は高々2倍程度である。

(3) 2レベルマイクロプログラム制御方式

1 マイクロ命令で起動されるナノプログラムのステップ数は、数値計算では7~70ステップと長く、言語処理では他のPUでの演算結果が頻繁に必要なので、1.1~1.5ステップ程度である。2レベルマイクロプログラム制御方式を利用した記憶量の最適化[9]を、MSDLインタプリタのような大規模なマイクロプログラムに適用すると、ナノプログラムメモリの割付率が90%近くになり、ナノプログラムメモリの使用効率が極めて良いことが判明した。

以上のように設計方針の観点からアーキテクチャの評価を行った後、MUNAPの命令セットアーキテクチャの評価を行った[14]。すなわち、命令セットを直交性、一様性、拡張性の観点から評価するための手法を考案し、MUNAPのマイクロ・ナノ命令セットに適用して評価した。この結果、2、3の命令の改善の可能性が指摘された。また、命令の動的遷移パターンを計測して、命令合成の可能性についても検討した。

6. プロジェクトの成果、反省点と将来展望

6.1 成果

(1) アーキテクチャ研究へのインパクト

当初設けた2レベルマイクロプログラムによる並列処理の制御、あるいは種々の非数値処理演算機能の装備といった設計方針は、10年の研究に耐える内容をもっており、計算機アーキテクチャの研究にもそれなりのインパクトを与えたものと思う。

(2) 実験・実証的な研究姿勢

本研究に於て、我々は、シミュレーション等による評価を行わず、一貫して、実現してどうだったということを実証的に明らかにしてきた。実際にやってみて明らかになった点が多々ある。各種応用を通してMUNAPの有効性を実証的に評価したことは意義深い。

(3) ハードウェア・ソフトウェア・応用の垣根を越えた幅広い研究

ハードウェアの設計と試作、基本ソフトウェアの作成、各種応用からアーキテクチャの評価まで一貫して研究できた。この辺りは自作機の強みであり、これらの垣根を取り払うことにより新たに生まれた研究の芽も多い。

(4) 研究・教育上の成果

限られたマンパワー（教官3名と修士課程大学院生・学部生）の中で、参考文献に上げたような多くの成果を出せた。この他、学位論文1、修士論文7などがある。プロジェクトに関係した学生は、ICの塊と格闘する中から、本からは得られない多くのものを学んでいったものと思う。

6.2 反省点

(1) ハードウェアの保守

ハードウェアにおいても、要求定義、システム設計、論理設計、実装設計、ラッピング、テスト、運用保守のライフサイクルが存在する。運用保守については、応用段階に入ってから、マイクロ診断の必要性を感じた。保守のためのマイクロ診断システムをハードウェア完成後直ちに作成しておくべきであった。

(2) マイクロプログラム作成支援システム

全ての応用は本システムを用いて作成するので、十分なテストと機能の改善が必要であった。アセンブラは絶対型コードを生成するので、再配置の必要性からローダを作成した。大規模マイクロプログラムの作成を考慮すると、最初から再配置型コードを生成するアセンブラと再配置・最適化を行うローダとした方が良かった。デバグも初期の使用の後、ブレーク実行を追加した。大規模マイクロプログラムの作成とシステムの評価を考慮すると、アセンブラ、ローダ、デバグ、エミュレータ、オプティマイザの体系を十分に検討し、信頼性の高いシステムとして実現しておくことが重要である。

(3) 応用分野の検討

言語処理においては非数値処理指向の効果は認められたが、並列処理の効果は高々2倍程度であった。並列処理の効果は数値計算において最も認められたが、応用研究の4年目からである。並列処理に適した問題をもっと探し、より早期に立ち上げるべきであった。

(4) 学生へのアーキテクチャ、システム使用法等の教育

学生の研究室への配属後、アーキテクチャと基本ソフトの使用法の説明、及び簡単なマイクロプログラムの作成に毎年、約1カ月を要した。アーキテクチャ教育、及び基本ソフトの使用法を支援するCAIシステムがあれば有効である。

計算機の試作・保守に当たって、ハードウェアに不慣れな学生によるトラブルも頻発している。例えば、ハンダ付の不良による誤動作、動作調整時の短絡によるIC等の破壊などが挙げられる。ハードウェア技術に関する基礎的な訓練も十分に行っておく必要がある。

(5) 仕様書の完備

システムの作成時、あるいは作成後に外部・内部仕様書を作成し、学生が入れ替わっても引き継げるようにした。仕様書を完備し、必要に応じて修正することは、研究を長期にわたって続ける上で極めて重要であった。今後は計算機による仕様書の管理が必要であろう。

(6) その他

論理設計からハードウェアの完成、基本ソフトの作成までにほぼ4年を要した。これが長いか短いかは分からないが、マシンの新規性を主張するためには、この期間をできるだけ短縮することが重要である。この期間に作成したシステムの善し悪しが応用システムの開発効率を左右するので、成果を焦らず腰をすえて、かつ短期間に作成する必要があろう。

6. 3 将来展望

(1) ハードウェアのVLSI化

プロセッサユニット、マイクロシーケンサなどのVLSI化を図ると、プロセッサがn台のMUNAP、あるいはマルチMUNAP(MUNAPがn台)などの実現が可能になる。並列処理の粒度の問題、並列処理の階層化など解決すべき問題は山積しているので、VLSI化により、さらに進んだ問題に挑戦できるであろう。

(2) 並列処理の記述

MUNAPにおいては、並列処理の記述は2レベルマイクロプログラムの記述の過程で行われた。これによって、細粒度の並列処理の可能性を目一杯引き出すことには成功しているが、大規模の並列処理問題をこのような言語で記述・作成するのは容易ではない。本研究におけるシステム記述言語MSDLは一つの方向を示したものであるが、更に高度の並列処理を考慮すると、個々の並列処理計算機の特長を損なうことなく、効率よく並列処理を記述できるような言語の設計が望まれる。

(3) 応用問題の選定

MUNAPではデータベースのように当初から対象とした応用もあったが、ユニバーサルホスト計算機があるからこれもやってみようというボトムアップ的な発想が多かった。現在、超/高並列処理マシンの研究が進められているが、並列処理の効果を上げるためには、まず適切な問題を決めてからマシンを設計するというトップダウン的な思考が重要となろう。

7. むすび

実験研究用の並列処理計算機MUNAPの開発から、応用・評価について述べた。さらに、研究プロジェクトの収束期を迎えての感想、将来展望などに触れた。

良かれ悪しかれ、10年余を一つの計算機とつき合うことになり、システムの製作者である我々も知らず知らずの内にMUNAPより多大の影響を受けている。それは試作や応用の研究過程を通じて実感として得たものであり、通常の論文ではそのようなことに触れる余裕はなかった。従って、本稿ではできるだけそういう面を掘り起こして記述するように心がけたつもりである。同様なプロジェクトを志す方に少しでもお役にたてば幸いである。

謝辞

MUNAPの開発初期の段階でプロジェクトの推進にご協力頂いた本学教育学部石川賢助教授、ソニー(株)ワークステーション事業部小林広幸氏(当時電通大学院生)、ならびにMUNAPの開発・応用に関わった多数の大学院生・学部学生・留学生に感謝する。

京都大学萩原宏教授、柴山潔助教授、新実治男助手、九州大学富田眞治教授には、貴重なご助言を頂いている。東芝総研小柳滋主任研究員、日本電気箱崎勝也システムインタフェース技術本部長(当時C&Cシステム研究所)、日立研究所坂東忠秋部長には、アーキテクチャ設計時に種々ご意見を頂いた。MUNAPのきょう体の製作は、工学部機械工場において行われたものである。試作に当たっては、文部省科研費の補助を頂いた。ここに記して謝意を表したい。

参考文献

論文誌あるいは国際会議録に採録となったものを

中心に年代順に並べた(ただし、[17]はプロジェクト全体についての著書)。

- [1]T. Baba, K. Ishikawa, K. Okuda, and H. Kobayashi: MUNAP - A Two-Level Microprogrammed Multiprocessor Architecture for Nonnumeric Processing, Proc. IFIP 8th World Computer Congress, pp.169~174 (1980.10).
- [2]馬場, 石川, 奥田: 2レベルマイクロプログラム制御計算機MUNAPのアーキテクチャ, 信学会論文誌, J64-D, 6, pp.518~525 (1981.6).
- [3]馬場, 石川, 奥田: 2レベルマイクロプログラム制御計算機MUNAPにおける非数値処理, 信学会論文誌, J64-D, 6, pp.526~533 (1981.6).
- [4]馬場, 橋本, 山崎, 奥田: 2レベルマイクロプログラム制御計算機MUNAPにおけるマイクロプログラムの記述とその処理, 信学会論文誌, J65-D, 10, pp.1265~1272 (1982.10).
- [5]T. Baba, K. Ishikawa, and K. Okuda: A Two-Level Microprogrammed Multiprocessor Computer with Nonnumeric Functions, IEEE Transactions on Computers Vol.C-31, No.12, pp.1141~1156 (1982.12).
- [6]T. Baba, K. Yamazaki, N. Hashimoto, H. Kanai, K. Okuda, and K. Hashimoto: Hierarchical Micro-Architectures of a Two-Level Microprogrammed Multiprocessor Computer, Proc. 1983 Int. Conf. on Parallel Processing, pp.478~485 (1983.8).
- [7]T. Baba, K. Yamazaki, N. Hashimoto, H. Kanai, K. Okuda, and K. Hashimoto: Experimentation with a Two-Level Microprogrammed Multiprocessor Computer, Proc. 16th Annual Microprogramming Workshop, pp.47~54 (1983.10).
- [8]山崎, 金井, 馬場, 奥田, 橋本(信), 橋本(和): 2レベルマイクロプログラム制御計算機MUNAPのシステム記述言語MSDLとその処理, 信学会論文誌, J67-D, 1, pp.149~156 (1984.1).

- [9]T. Baba, M. Ikeda, K. Yamazaki, and K. Okuda: Compaction of Two-Level Microprograms for a Multiprocessor Computer, Proc. 17th Annual Microprogramming Workshop, pp.95~103 (1984.10).
- [10]山崎, 金井, 土井, 小野, 馬場, 奥田: システム記述言語MSDLの処理系におけるタグ付きアーキテクチャ, 信学会論文誌, J67-D,10, pp.1202~1209 (1984.10).
- [11]大谷, 馬場, 稲川, 岩崎, 奥田, 山崎: 抽象データ型による関係データベースシステムの間合せ処理, 信学会論文誌, J67-D,12, pp.1450~1457 (1984.12).
- [12]山崎, 金井, 馬場, 奥田: ユニバーサルホスト計算機MUNAPのアーキテクチャの評価, 信学会論文誌, J69-D,1, pp.21~29 (1986.1).
- [13]馬場, 佐野, 鈴木, 山崎, 奥田: 計算機アーキテクチャの支援によるソフトウェアテストシステム, 信学会論文誌, J69-D,1, pp.30~41 (1986.1).
- [14]K. Yamazaki, T. Baba, K. Okuda, and H. Kanai: Architectural Evaluation and Improvement of a Universal Host Computer MUNAP, Proc. IFIP 10th World Computer Congress, pp.779~784 (1986.9).
- [15]稲川, 馬場, 石川, 山崎, 奥田: P r o l o g の単一化における引数間の並列処理, 信学会論文誌, J70-D,2, pp.298~306 (1987.2).
- [16]土井, 山崎, 大川, 馬場, 奥田: 低レベル並列処理計算機MUNAPによるオブジェクト指向型言語の高速処理, 信学会論文誌, J70-D,2, pp.307~314 (1987.2).
- [17]T. Baba: Microprogrammable Parallel Computer-MUNAP and Its Applications, The MIT Press, Computer Systems Series, p.290 (1987.6).
- [18]K. Yamazaki, S.Ookawa, T. Baba, and K. Okuda: Low- and High-Level Parallelism for an Object-Oriented Language, Proc. 3rd Int. Conf. on Supercomputing, pp.383-390 (1988.5).
- [19]馬場, 加賀谷, 吉永, 鈴木, 山崎, 奥田: MUNAP上の拡張L⁶言語を用いた3次元色彩図形処理, 信学会論文誌(採録決定).
- [20]山崎, 松島, 奥田, 馬場: 知識工学的手法に基づいたマイクロ診断システム, 信学会論文誌(投稿中).