

# 第1章 緒論

## § 1.1 本論文の背景および目的

1940年代の後半に最初の真空管式電子計算機が登場して以来、計算機は様々な改良を受け、その性能を飛躍的に向上させた。例えば、使用される電子回路素子は真空管からトランジスタを経て、現在は目的に応じて様々な速度および集積度を持つ集積回路が使用されるように変化してきた。これによって計算機の手度は向上し、また同程度の性能であれば小型化されるようになった。このようなハードウェアの進歩とともに、使用される計算機用言語やOSといったソフトウェアも改良されてきた。そしてこのハードウェアとソフトウェアの相互関係の中で、様々な構成を持った計算機が開発され、その中には広く使用されるようになったものもあり、またあるものは歴史の中に消えていった。

現在広く使用されているいわゆるノイマン型のプログラム内蔵型計算機は、高速動作可能なデバイスの発達やパイプライン方式をはじめとした各種のハードウェア構成の改良により、これまでは着実にその性能を向上させてきた。しかしながら、近年ではハードウェアを実現する際の各種デバイスの物理的な性能限界のため、単一の処理装置を用いた逐次処理では処理性能向上の限界に近づきつつあるといわれている。従って、今後のさらなる処理性能向上のためには、何らかの並列処理の導入が必要不可欠である。

並列処理方式には、並列に処理される対象の大きさ（粒度）によって、命令レベルの並列性を利用するものから、多数の計算機をネットワークで結合し並列動作させるものまで幅広い構成方式がとられ、多くの研究が進められている。その中でも特に近年研究や実用化が進められている手法として、命令レベルの並列処理方式が多数提案されており、現在ではパーソナルコンピュータレベルのCPUにも使用されるようになっている。命令レベルの並列処理方式が一般化した理由としては、より粒度の大きな他の並列処理方式とは異なり、プログラミングの基本的な構成方法に大きな変更を加える必要が無く、これまでの方法で作成してきたプログラムはそのまま高性能化をはかることができるという利点による。一方、超並列計算機システムと呼ばれる多数のプロセッサをネットワークで結合したシステムも実用化に向

けた研究開発が進んでおり、近年では商用のマシンも発表されている。しかしながら、これらの超並列計算機システムでは、その性能を最大限に発揮するためには、プログラミングの考え方を並列処理方式に合わせて大きく変更しなければならないことが一般的である。従って、すでに存在する応用プログラムをそのまま使用したのでは高性能を発揮することは不可能であり、また、現状では超並列システム用のプログラムの開発、デバッグは困難な場合が多く、使い勝手に問題が残っている。

これまでに研究・開発が行われてきた命令レベル並列処理方式アーキテクチャは多数あるが、その中でも代表的なものとしては、スーパースカラ(Superscalar) [Agewala 87;Johnson 91;Motorola 91]と VLIW[Charlesworth 81;Colwell 88;Ellis 86;Fisher 83,84]があげられる。

スーパースカラ・アーキテクチャは、各命令間の依存関係を検知するハードウェアを用いて、並列に動作可能な命令を選択し、複数の処理ユニットを制御して命令レベル並列処理を行う方式である。

一方、これとは対照的なアーキテクチャとしてVLIWがある。VLIWアーキテクチャは多数の並列動作可能な処理ユニットを、コンパイラにより静的に構成された一つの語長の非常に長い命令により、それぞれ独立かつ直接に制御することにより並列処理を行う方法である。

これらの命令レベルの並列処理方式以外で、現在広く使用されている方法の一つに、ベクトル処理方式がある [Jippo 87;Karaki 84;Yasumura 85]。これはいわゆるスーパーコンピュータで使用されている方式で、配列のようなデータ構造を効率よくパイプライン方式の演算器で処理することにより、特に数値演算の分野で高性能を発揮する方式である。

そこで、本研究では、21世紀に向けて様々な応用分野においてますます要求されるであろう計算機の処理能力に対する要求を満足すべく、現在広く使用されているこれらの3つの手法を統合し、それらの利点を生かした上で弱点を補い合うという発想の元に、新たな汎用高速命令レベル並列処理方式の方法論を確立することを主な目的とする。そのために、具体的なシステム構成、ソフトウェア環境、さらには応用例に至るまでの各種の事柄について考察を行う。

## § 1.2 本論文の内容

第1章は、緒論である。

第2章では、まず計算機を構成する各レベルや、計算機的设计プロセスと関連付けた新しい計算機アーキテクチャの定義を示す。次に、次章以降で述べる新たなアーキテクチャの基礎となるこれまでに提案・実現されてきた各種の命令レベル並列処理方式の中から、特に代表的なものであるスーパースカラ、VLIWといったアーキテクチャについてその長所、欠点を考察する。また、主に数値計算の高速化を目的として用いられるベクトル処理アーキテクチャについても検討する。

第3章では、第2章での検討に基づき、現在よりも高速かつ汎用なCPUの実現を目的としたアーキテクチャである、ジェットパイプライン(Jetpipeline)・アーキテクチャを提案する。まずこれまでの命令レベル並列処理アーキテクチャとその問題点について述べ、それを解決するためのアーキテクチャについて、まず基本概念を示し、次にハードウェア、ソフトウェアの両面について具体化する。

第4章では、提案したアーキテクチャについて、シミュレーションによる性能評価を行う。シミュレーションの際に仮定したハードウェア構成、および各種パラメータについて示した後、シミュレーション結果を示し、ベンチマークプログラムの性質等を考慮しアーキテクチャの性能について検討を行う。

第5章では、第3章、第4章で提案・評価したジェットパイプライン・アーキテクチャが高性能を発揮できるであろう応用の一例として、現在各方面において広く用いられるようになってきているコンピュータグラフィックスの並列処理システムに用いた例について、システム構成例とシミュレーションによる性能評価を行う。ここで使用する画像生成アルゴリズムの性質から、ジェットパイプライン・アーキテクチャにおけるVLIW的並列化が有効であることを明らかにする。さらにシステムレベルの並列化を加えた2段階の並列処理を行うことにより、より一層の高速化が実現可能であることを示す。

第6章は結論である。