第 6 章　結論

本論文では、近い将来に訪れるであろうマルチメディア・ネットワーク社会において、ますます需要が高まると思われる、数値・非数値の両方の応用に対して優れた性能を発揮可能な計算機アーキテクチャの進化の一形態として、現在使用されつつある数種の命令レベル並列処理方式とベクトル処理方式を統合したジェットパイプライン・アーキテクチャを提案し、その概念、構成、性能評価、および応用例を示すことにより、新たな命令レベル並列処理アーキテクチャについて一つの方向を示した。以下で、本論文の内容を総括する。

第 2 章では、まず階層化された計算機の構成とその設計プロセスに基づいて、新しい計算機アーキテクチャの定義を示した。続いて、現在の高性能計算機アーキテクチャの基本となっているRISCアーキテクチャについて述べ、近づきつつあるその性能の限界を解決するために用いられてきているいくつかの代表的な命令レベル並列処理方式と、スーパーコンピュータにおいて数値計算処理の高速化のために広く用いられているベクトル処理方式について実例を挙げながら述べ、その長所および欠点について検討した。

第 3 章では、第 2 章で検討した各種の命令レベル並列処理方式とベクトル処理方式を統合し、汎用かつ高性能な計算機アーキテクチャをめざしたジェットパイプライン・アーキテクチャを提案し、その概念を示した。ジェットパイプライン・アーキテクチャでは、スーパースカラのように複数の命令を並列に実行するが、ハードウエアの複雑化を抑えるために命令の並列化やスケジューリングはVLIWのようにソフトウエア（コンパイラ）で行う方式をとっている。このような方式を採用することにより、ハードウエア量をできるだけ抑えながら柔軟な命令レベル並列処理を実行可能となっている。また、ベクトル処理機能をも導入することによって、数値演算処理性能の向上を図っている。次に、このような概念に基づいた具体的なハードウエア構成を各機能ブロックごとに示した。とくに、スカラレジスタにおいてはハードウエア量の爆発を防ぎ、アクセスの自由度もできるだけ大きくなるように考慮された、バンク化オーバラップレジスタ構成を提案した。ソフトウエア環境についても、命令セットや命令のスケジューリングなどについて検討した。特にソフトウエア環境においては、通常の逐次的なプログラムから並列性を引き出し、ジェッ

トパイプライン・アーキテクチャ上に適切にスケジューリングするための方法について、基本的な手法、そして高度な命令レベル並列処理を実現できるディスパッチスタックやソフトウエアパイプライニングといった手法のジェットパイプラインへの適用法について検討した。特に、ソフトウエアパイプライニングを用いる場合に、ループの各イタレーションが並列に実行されるが、この際に本論文で提案したジェットパイプライン・アーキテクチャのバンク化オーバラップレジスタを適切に利用したレジスタ割り当てを行うことにより、効率よく実行できることを示した。

第4章では、第3章で提案したジェットパイプライン・アーキテクチャについて、その性能を評価するためにシミュレータを構築し、シミュレーションによる性能評価を行った。数値計算を主にしたリバモアループと、非数値計算処理を主とするスタンフォードベンチマークの2種類のベンチマークプログラムを実行しジェットパイプラインを評価した。リバモアループには、ループのイタレーション間に依存関係がなく並列化が可能なものから、イタレーション間に依存関係があり並列化が困難なものまで各種のループが含まれている。このベンチマークに対しては、スカラ命令を用いた場合にはディスパッチスタックによる並列化とソフトウエアパイプライニングによるものとの両方をシミュレータ上で実行し、逐次処理を行った場合との比較を行った。この場合、特に並列化が可能なループについてはソフトウエアパイプライニングを用いることにより理想値の4に近いスピードアップが得られた。また、並列化が困難なループについても、ソフトウエアパイプライニングを用いることにより高速化が可能であった。スタンフォードベンチマークはリバモアループのような単純なループのみから構成されてはおらず、関数呼び出しや再帰呼び出しといった通常のプログラムに近い処理を含んでいる。そのため、このベンチマークについてはスカラ命令のみを用い、ディスパッチスタックを用いた並列化により評価した。その結果、リバモアループよりは若干落ちるものの、最大で2倍強の高速化が得られた。最後に、リバモアループに対してベクトル処理を適用した場合について評価した。この場合、ベクトル化が容易なものについては高速化が達成できたが、ベクトル化が困難なものについては他の手法による並列化の方がよりよいスピードアップが得られた場合があった。これらのことより、従来の方法で高速化可能な処理ばかりでなく、従来は高速化困難であった処理に関しても並列化の手段を適切に選択して適用することにより高速化が可能であるということが示唆された。

第5章では、ジェットパイプライン・アーキテクチャの一つの応用として考えられる、コンピュータグラフィックスへの応用について、光線追跡法による画像生成の高速化のための並列処理システムのハードウエアおよびソフトウエアについて検討した。まず、光線追跡法を用いた動画像生成を高速に行えるアルゴリズムである部分更新光線追跡法について、その膨大な記憶容量を低減し、また再計算が必要な光線を高速に発見できるアルゴリズムを新たに提案し、実際に動画像の生成を行い、その性能を評価した。その結果、2フレーム目以降の再計算においては、従来の数倍から十数倍の高速化が達成できた。また、本アルゴリズムを高速に実行するのに適している粗結合マルチプロセッサシステムによる並列処理についても、シミュレーションと実機による実験の双方から評価した。ただし、実機における実験では密結合型の並列処理システムを使用したため、主に負荷分散の効果について評価した。その結果、負荷分散を考慮した処理の割り付け、具体的には各処理ユニットが担当する画面上の画素の割り当てを工夫することにより、かなり効率の良い並列処理を実現でき、結果として高速に動画像を生成可能なことを明らかにした。また、マルチプロセッサシステムのCPUとしてジェットパイプライン・アーキテクチャを用いた場合に可能となる、画像生成アルゴリズム内に存在する処理の対称性に基づいた命令レベル並列性の利用による高速化の可能性についても検討した。これらのことから、ジェットパイプライン・アーキテクチャにおける命令レベル並列処理とプロセッサレベルの並列処理を組み合わせることにより、高速な画像生成が可能であるという結果が得られた。

以上より、ジェットパイプライン・アーキテクチャはリバモアループのような数値計算への応用分野に対しても、またスタンフォードベンチマーク中のハノイの塔やソートといった非数値計算的な応用分野においても、単一のプロセッサとして高性能を発揮することができ、またマルチプロセッサシステムのようなより大粒度の並列処理システムの構成要素として用いることによってシステム全体の性能を向上させることが可能であると思われる。

今後の課題としては、

・より広範囲なプログラムによる性能評価
・ベクトル処理と命令レベル並列処理との分担に関する評価
・最適化コンパイラの開発

・メモリシステムの詳細化とそれによる性能への影響の評価
・実用の際必要となるであろう、仮想記憶およびOSのサポート用ハードウエアの付加
など、数多くの問題が残されている。