

# インテル次世代64ビット型 プロセッサIA-64のアーキテクチャ

中村 正澄  
インテル (株)

IA-64はEPIC (Explicit Parallel Instruction Computing) と呼ぶ新しいテクノロジーを取り入れたマイクロプロセッサアーキテクチャである。インテルではi386™プロセッサから始まった32ビットマイクロプロセッサをIA-32プロセッサ、EPICを採用した64ビットマイクロプロセッサをIA-64プロセッサと呼ぶ。Merced™プロセッサは、IA-64アーキテクチャの最初のプロセッサである。Mercedプロセッサの市場への量産出荷を2000年半ばに予定している。Mercedプロセッサに限らず、すべてのIA-64プロセッサはユーザの既存の資産を保護するために、IA-32プロセッサとバイナリレベルでの100%互換性が確保されるハードウェア機構を持っている。また、さらに周波数を高めた製品McKinley (開発コード名)を2001年後半に、ならびに高性能を追求したMadison (開発コード名)、コストパフォーマンスを追求したDeerfield (開発コード名)を計画している (図-1)。

本稿では、Mercedプロセッサの概要ならびにIA-64プロセッサアーキテクチャの概要を紹介するとともに、Mercedプロセッサにおけるインプリメンテーションについて述べる。

## インストラクションセットアーキテクチャ

EPICとは、新しい技術手法である。これまでもマイクロプロセッサの処理能力を高めるために、CISC、RISC、スーパースケーラ、スーパーパイプラインなど、数々のプロセッサアーキテクチャが発表されてきたが、EPICはその次世代を担う新しいアーキテクチャである (図-2)。この新しいアーキテクチャは、インテルとヒュレット・パッカード社が1994年6月に研究プロジェクトを発足し、開発してきたものである。

EPICの特徴を簡単にまとめると、静的な並列化をコンパイラがスケジューリングしハードウェアによるリオーダーによるコストを抑えることができるアーキテクチャである。

図-3にIA-64のインストラクション・フォーマットを示した。EPICの大きな特徴の1つは命令レベルでの並列化である。プロセッサはバンドル単位で実行されていく。バンドルは128ビット幅で、3つのインストラクション (命令) とテンプレートから構成される。デコーダは、テンプレートを参照することで、インストラク

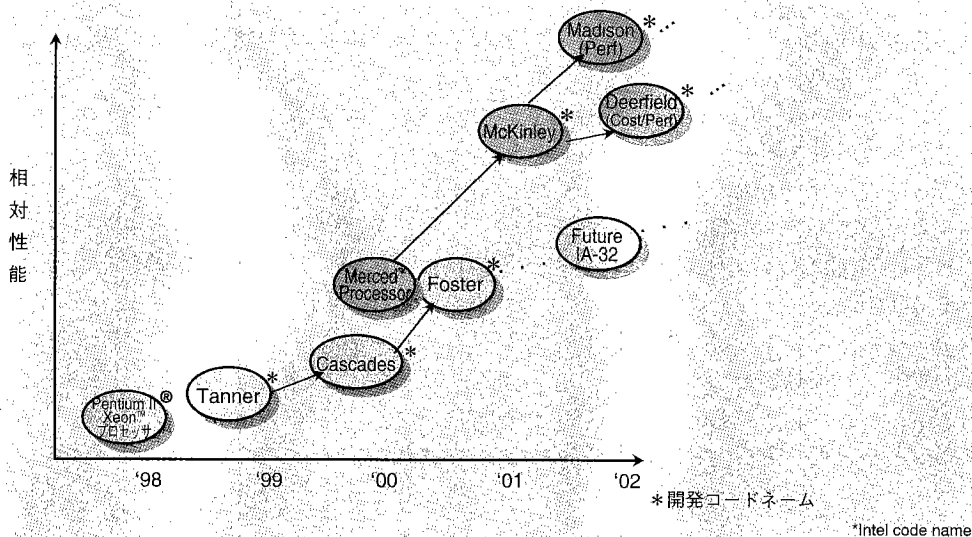


図-1 マイクロプロセッサロードマップ

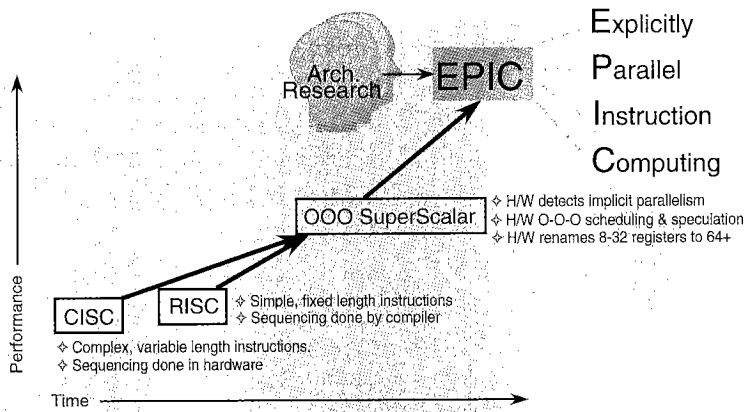


図-2 次世代アーキテクチャ技術

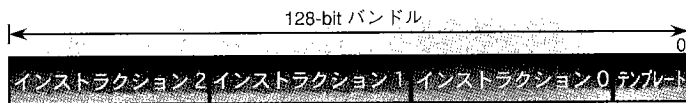
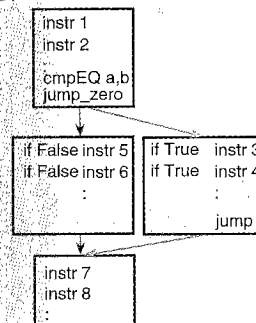


図-3 IA-64 インストラクションフォーマット

ションの組合せを知ることができる。命令はプロセッサに送られる時点ですでに並列化されているのである。IA-64アーキテクチャでは、性能を向上させるためにハードウェアの役目はクロックあたりに実行できるバンドル数を上げていくことにある。

IA-64アーキテクチャは、この新しいEPICを取り入れるとともにIA-32マイクロプロセッサの100%互換を保っている。数式で表すとIA-64=EPIC+IA-32となる。

従来のアーキテクチャ



EPIC アーキテクチャ

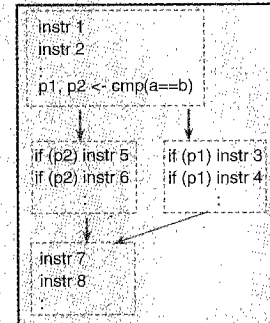
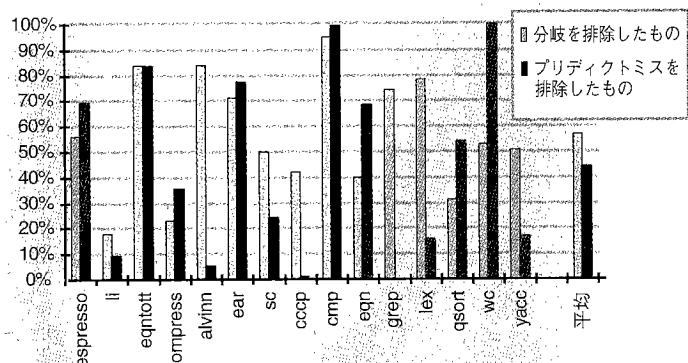


図-4 ブリディケーションの概念図

### プロセッサ性能を妨げる要因

周知なことであるが、一般的にプロセッサの性能を妨げる要因の1つに分岐を挙げることができる。従来のプロセッサでは、パイプラインの乱れを防ぐために分岐を予測し、命令の読み込みから実行までの流れを止めない努力を行ってきた。分岐予測のアルゴリズムや機能を向上させることでほとんどの分岐による乱れを防ぐことができるようになってきた。しかし、現実にはまだ分岐によるパイプラインの乱れは存在し続けている。特に予測が外れたときのペナルティは非常に大きい。周波数を上げるために、パイプラインが深くなればなるほどペナルティは大きくなる。分岐予測が外れた際の性能ペナルティは20～30%に達することがインテル社内の研究で判明している。

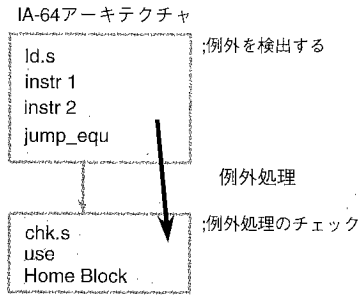
また、分岐はコンパイラにとってプログラ



米イリノイ大学の研究によると、彼らのHypothetical eight-wideシステムでは分岐を排除するよりブリディクトミスをなくす方が性能が向上するというレポートが報告されている。

出典: International Symposium on Computer Architecture '95 S.Mahlke, et.al.

図-5 ブリディケーションによる性能向上



データの投機実行によって、事前に発生しうるページフォルトなどの例外処理を事前に処理させることができる。

図-6 投機実行の例

ムを小さなブロック単位に分解してしまい、命令レベルの並列化や再配列による並列実行可能な部分を見つけ出す作業を困難にしてしまう。IA-64アーキテクチャでは、この問題解決に対しプリディケーション (Prediction) を提供する。図-4にプリディケーションの概念図を示した。プリディケーションとは、分岐先をいかにうまく予測し流れを止めずに実行するのではなく、分岐先をすべて実行してしてしまうというアイデアである。

IA-64アーキテクチャでは、通常のレジスタとは別に条件が成立したか、不成立であったかを示すプリディケートレジスタを64個持っている。レジスタ幅は1ビットである。プリディケートによって条件分岐に対し、分岐条件が成立するまで結果を待たずに先の命令実行を続けることができる。アセンブリ言語で記述する場合プリディケートレジスタは、インストラクションの先頭にタグのように記述し、条件の成立、不成立によって実行されるものとされないものを明示する。条件が不成立だった演算結果はプロセッサ内で捨てられる。レジスタやメモリへの書き込みは行われない。また、実行前の演算はプロセッサが実行されず、リソースを無駄に使用しない。プリディケートレジスタをインストラクションに割り振るのは、コンパイラの役目であり、プリディケートレジスタを使用して、演算が正しく行われるようコントロールするのはプロセッサの役目である。図-5に米国イリノイ大学のレポートの一部を示した。プリディケーションの有効性を裏付ける興味深いシミュレーション結果がシンポジウムで報告されている。

性能を妨げる2つ目の別な要因として、メモリへのレイテンシ (遅延) が上げられる。インテルのプロセッサ性能は18カ月で約2倍に向上しているが、メモリとの物理的な距離、ならびにスピードの距離 (アクセスタイム) は広がる一方である。キャッシュは、そのメモリアクセスの遅延を隠す技術であるが、プロセッサの速度が上がれば上がるほど、同じシリコン上の最も近いレベルのキャッシュでさえデータの読み出しに数クロック必要になってきている。今後、メモリとの距離は

さらに広がり、キャッシュへのアクセス・レイテンシは無視できないほど大きくなっていくことが予想される。もし、メモリ・レイテンシの問題が解消されなければ、並列化のために用意された命令実行スロットは、多くが空きスロットになってしまう懸念がある。同時実行できる命令が大きくなればなるほど、メモリ・レイテンシは大きな障害となってくる。

IA-64アーキテクチャでは、投機実行 (Speculation) の機能でこの問題を解決する。遅いメモリに対し事前に必要となる (必要でないかもしれない) メモリを参照するようハードウェアに要求する命令がコンパイラに与えられる。

演算は、当然レジスタを使用して行われる。演算に必要なデータがメモリにあれば、メモリを読みに行かねばならない。投機実行とは、メモリを参照しなければならないロード命令を演算に必要な前にプロセッサに読み込むことで、メモリ参照にかかる時間を見えなくする技術である。コンパイラは、事前に必要となるデータをプロセッサに読み込ませる (レジスタに格納する) という簡単な作業、命令を発行することができる。

図-6にメモリ参照によるページフォルトが発生するケースを示した。コンパイラは、スペキュレーティブ・ロード命令を使用し事前に必要となるデータをメモリから読み出すよう指示することで、ハードウェアは、その必要されるメモリアクセスがページフォルトを起こすか事前に調べることができる。また、事前に読み出されたメモリは、プロセッサに取り込まれ、必要になったときにすぐに提供することができる。チェック命令は、例外処理が終了しているか、もう必要なデータを使用することができるのか調べるチェック・ポイントとなる。もし中間処理が終了していれば、そのまま次の命令でレジスタに取り込まれたデータを使用することができる。

データを書き込む場合 (ストア) も同様である。インテルのプロセッサでキャッシュがライトバックをサポートするようになったのはi486™プロセッサからだが、ライトバック・キャッシュをサポートしていれば事前にキャッシュにメモリの内容を読み込むことで、外部メモリへのアクセスを行わずに演算結果をすばやく更新することができるのである。IA-64アーキテクチャではキャッシュへのデータ取り込み命令をサポートする。

性能を妨げる3つ目の要因は、プログラムがシーケンシャルに実行されることを前提としたマシンコードで実行されることにある。従来のプロセッサでは、古くから使われてきたインストラクション、つまり命令をフェッチし、実行し、また命令をフェッチし、実行する、といった古いインストラクションでマシンコードが構成されている。コンパイラはソース・プログラムを解析し並列実行できる部分を見つけ出す。並列実行

させることで実行速度がクリティカルだったパスを最適化させる。最近のコンパイラでは、並列実行度を高めるために、プログラムを変化させることまで行っている。しかし、コンパイラがすべての作業が終了した後、シーケンシャルなマシンコードが発行されていくため、いくらかの疑似的な連続性が入りこむ機会を与えてしまう。そのため、プロセッサは再びハードウェアで並列性を探し、独立した命令を並列実行していく。完全なものを追求すればするほど、このために必要なハードウェア資源は、膨大なものとなる。

それよりも、コンパイラにアプリケーションをどのように最適化し性能向上させていくか任せの方がより効果的である。このように考えられて作られた技術手法がEPICであり、IA-64アーキテクチャである。

## インプリメンテーション

EPICを有効に動作させるためには、多くのレジスタセットが必要となる。多くのエグゼキューションユニットを止めることなく動作させる必要があるためである。IA-64アーキテクチャでは128の整数レジスタ、128の浮動小数点レジスタが用意される。またMercedプロセッサでは、レベル0データキャッシュは整数レジスタ群の直近に位置し高周波以下においても少ないレイテンシでアクセスできるよう設計されている。これは、プロセッサの周波数が高くなるにしたがって同じシリコン上のキャッシュであっても、アクセスレイテンシを数クロック内に抑えることが難しくなっているからである。

Mercedプロセッサのインプリメンテーションでは、整数および浮動小数点演算のユニットが複数ユニット用意されている(図-7、図-8)。各演算ユニットは複数のリード・ライトポートを通じて128個のレジスタファイルに結合されている。

整数実行ユニット(ALU)は2つのロード・ストアと複数のリードライトをサポートすることができる。またALUはレイテンシを抑えるためバイパスメカニズムを持っている。

浮動小数点ユニットは4つある。2つの倍精度浮動小数点演算ユニットと2つの単精度浮動小数点ユニットから構成され、各ユニットは同時に演算を行うことができる。最大で8つの単精度浮動小数点演算、または4つの倍精度浮動小数点演算をサイクルごとに実行することができる。各ユニットにはマルチポートでレジスタファイルと結合されており、演算を止めることなく次々と処理するデータを送ることができる。1999年前

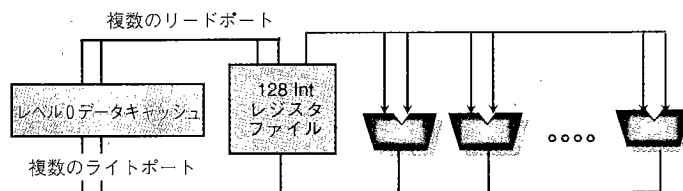


図-7 整数演算ユニットブロック図

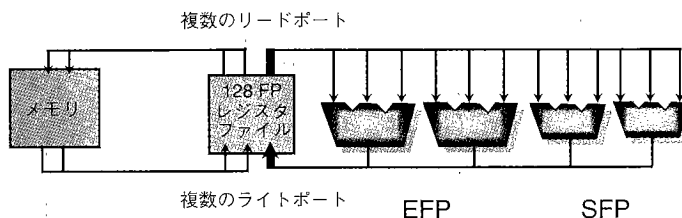


図-8 浮動小数点演算ユニットブロック図

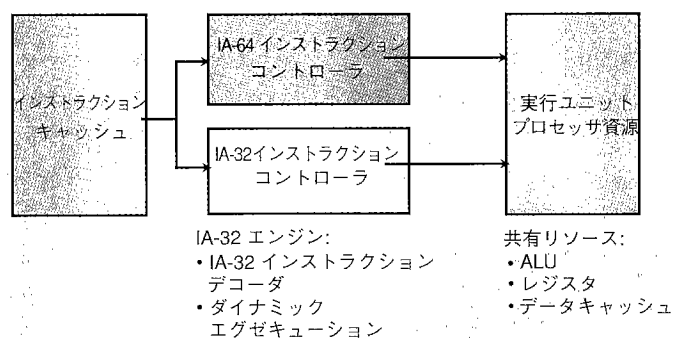


図-9 IA-32互換性

半に製品化が予定されているIA-32プロセッサKatmaiと同様のSIMD (Single Instruction Multiple Data) インストラクションもサポートする。浮動小数点演算ユニットはPentium®Proプロセッサの約20倍、1999年発表予定のTannerプロセッサの約3倍以上の3Dグラフィック性能が見込まれている。

## IA-32アーキテクチャとの互換性

IA-64アーキテクチャは、IA-32アーキテクチャと完全互換を保っていることは先に述べた。必要であれば、MS-DOSをブートさせることも可能である。

プロセッサはIA-32とIA-64バイナリの両方を実行するために別々の命令デコーダを持つ(図-9)。しかし、ALU、レジスタ、キャッシュなど、演算に必要なユニットはIA-64のリソースを共用し無駄なトランジスタ資源を浪費しない。IA-32命令は、IA-64のレジスタを使用し、IA-64の演算ユニットの中で実行される。

繰り返しになるが、IA-32のバイナリをそのまま実行することができるため、IA-64OSの配下でIA-64アプリケーションとIA-32アプリケーションを同時に利用することも可能である。

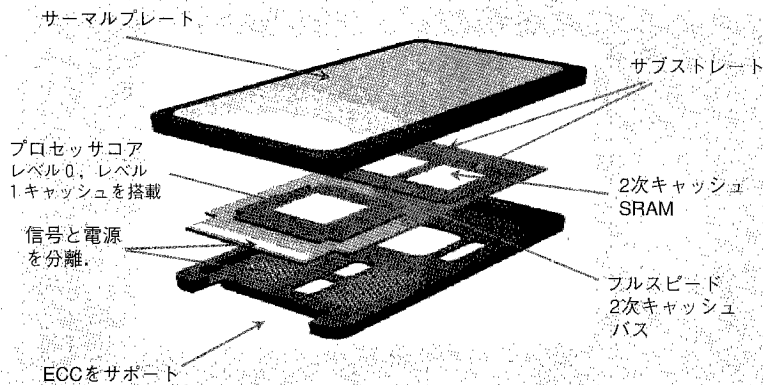


図-10 Mercedプロセッサカートリッジ

## Merced プロセッサ

Merced プロセッサは、0.18 ミクロンのCMOS プロセスで量産製造する。基幹サーバやハイエンドワークステーションなどの用途に向け、信頼性の向上に留意して設計されている。

図-10に示すようにMerced プロセッサは内部で複数のシリコンがMCM接続された、カートリッジ形状である。カートリッジ内に3つのレベルのキャッシュを搭載する。インストラクションとデータを区別したレベル0、大容量ユニファイドレベル1、サーバおよびワークステーションクラスで望まれる大規模のレベル2キャッシュを用意する。Merced プロセッサでは、レベル0とレベル1キャッシュはプロセッサと同じシリコン上に搭載され、大容量のレベル2キャッシュは別シリコンでカートリッジ上に搭載する。統合されるレベル2キャッシュSRAMも同様にインテルで設計製造され、プロセッサコアと同周波数で動作する。2次キャッシュは、外部バスと切り離された独立したバスを持ち、2次キャッシュへのアクセスが外部バスへのアクセスを妨げることがない。2次キャッシュのバスバンド幅は10GB以上である。今後、集積度が進んだ段階で2次キャッシュとプロセッサを同じシリコン上に集積する計画である。

プロセッサとキャッシュSRAMはオーガニックパッケージでカートリッジ内に封印される。カートリッジ内でMCM化される。

最近では、周波数とともに増大する電源、発熱がシステム設計者の問題となるが、Merced プロセッサではプロセッサとキャッシュ用の電源はピンから供給せず、電源専用のカードエッジを用意した。電源と信号を分けることによって、信号ピン周辺の高周波数動作するレイアウトに集中できるように工夫されている。カートリッジ下面のピンはすべて信号とGNDピンとなる。カードエッジには、プロセッサ専用の電源を供給

する。

また、発熱を有効に制御するために特別な新しいヒートプレートを開発した。従来のヒートシンクを用いた空冷でカートリッジ上面に取り付けられたヒートプレートを通じシリコン表面温度を適正な温度に制御することができる。またシリコン上に温度を監視する機能を取り入れた。外部から、風量をコントロールすることや、プロセッサが熱による誤動作を引き起こす前にシステムをシャットダウンすることなく運用することが可能になる。

信頼性の向上にも留意してされている。外部バスにECCをサポートすることはもちろん、カートリッジ内の2次キャッシュバス、シリコン内部のL1キャッシュにまでECCをサポートする。1ビットのソフトエラーをハードウェアで自動訂正でき、2ビットのエラーを検出することができる。マシンチェックアーキテクチャも拡張され検出されたエラーがどこで発生したものであるか細かく限定することができる。また、エラーによるデータの汚染をファームウェアやOSで訂正することや、特別なプロセッサ・ステートを経由することで、エラーのないプロセスに影響を及ぼさない。プロセッサ内部で発生したエラーでさえもシステムを停止することなく継続するリカバリー機能を提供することが可能になる。

## まとめ

インテルの64ビットマイクロプロセッサ、IA-64アーキテクチャは、新しい技術手法を取り入れ、従来性能向上の妨げとなった問題点を解決したマイクロプロセッサアーキテクチャである。Merced プロセッサは、最初のIA-64アーキテクチャ製品で、サンプル出荷を来年1999年に予定しており、量産出荷を翌年の2000年中頃に予定している。

(平成10年10月30日受付)