論 文 の 内 容 の 要 旨

論文題目　　プロセス微細化の影響を考慮した
非同期式プロセッサの構成に関する研究

氏 名　　小沢 基一

命令セットを固定した場合，プロセッサの性能は 1 サイクルで実行される命令数 IPC (Instructions Per Cycle) とクロック周波数の積で表せる．プロセッサにおける現在までの飛躍的な性能向上は IPC とクロック周波数の向上が同時に実現されてきたことによる．

まず，クロック周波数の向上はプロセスの微細化によりもたらされた．一般に，プロセスが微細化するとゲート遅延が減少する．これによりクリティカルパス遅延が減少し，クロック周波数の向上が実現される．

さらに，アーキテクチャの改良により IPC の向上がもたらされた．特に複数命令の同時実行を実現するスーパースカラ，命令の実行順序を動的に決定する out-of-order の採用が大幅な IPC の向上を可能とした．一般に，これらのアーキテクチャの改良はプロセッサの複雑化，大規模化をもたらすため，クロック周波数を決定するクリティカルパス遅延の増加が起こりうる．しかし，今まではクリティカルパス遅延の多くがゲート遅延に依存したため，この遅延増加をプロセス微細化によるゲート遅延減少で打ち消すことができた．そのうえ，回

路設計技術の進歩でさらに遅延が減少したため，プロセッサが複雑化，大規模化したにも関わらずクロック周波数の向上が実現された．

しかし，プロセスの微細化では，ゲート遅延は減少しても配線遅延は減少しない．そのため，今後は，遅延の多くが微細化で減少しない配線遅延となる．この場合，プロセス微細化による遅延減少は，長い配線に依存する構造ほど小さくなってしまい，チップ全体における大幅な遅延減少は期待できなくなる．この結果，クリティカルパスが長い配線で決定するようなアーキテクチャではクロック周波数の向上が制限される．

現行の高性能プロセッサの多くは out-of-order 実行を行う．このようなプロセッサのクリティカルパス遅延は，命令の実行順序を決める動的スケジューリング機構，分岐予測機構，キャッシュなどのメモリに基づく構造で決定される．一般に，メモリアクセス遅延の多くは配線遅延であるため，プロセス微細化による遅延減少はそれほど大きくなく，プロセス微細化によるクロック周波数の向上が制限される．

このような状況でクロック周波数を向上させるには，クリティカルパス遅延を決定する構造のパイプライン化が必要となる．しかし，これらの構造の多くは直前の出力を入力として利用するため，パイプライン化を行ってもストールが多く発生し，クロック周波数の向上による性能の向上が打ち消される．

このような場合でも，IPC を大幅に向上させることができれば，性能が向上する．しかし，一般のプログラムが持つ並列度は比較的小さいため，実現可能な IPC の上限も比較的小さい．さらに，現行の out-of-order アーキテクチャで IPC を向上させるには，メモリに基づく構造の容量やポート数を増加させる必要があり，配線長が増加する．その結果，クリティカルパスの遅延が増加し，前述したようなクロック周波数の制限が問題となる．

これらの結果，現行の out-of-order アーキテクチャを用いたプロセッサでは，プロセス微細化による IPC やクロック周波数の大幅な向上が期待できず，性能が頭打ちとなってしまう．

この問題は，現行の ouf-of-order アーキテクチャのクリティカルパスがプロセス微細化による遅延減少の小さい部分で決定されることによる．そのため，IPC を維持したままクリティカルパスをプロセス微細化による遅延減少が大きい部分に移動できれば，同時実行幅の増加などにより IPC を向上させても，クロック周波数の向上が期待でき，現在までの性能向上率を維持することが可能となる．そこで，本論文では，この条件を満たす Cascade ALU アーキテクチャを

提案する．

Cascade ALU アーキテクチャでは，RAW 依存を無視して命令を同時発行し，命令実行の際に ALU を直列接続して RAW 依存を解決する．一般のアプリケーションは RAW 依存を多く含むため，命令実行遅延は ALU 複数個の遅延となり，サイクルタイムをゲート遅延が支配的な ALU 遅延で制約できる．

Cascade ALU アーキテクチャでは，クリティカルパスの遅延が実行する命令列の依存関係で変動するため，非同期式論理を用いた実装が容易である．しかし，現行の非同期式論理を用いたプロセッサの性能は，同期式論理を用いた場合に比べて極端に低く，提案するアーキテクチャの実装には適さない．そこで，本論文では，パイプライン化を用いて非同期式論理に特有なオーバーヘッドを隠蔽する手法を示した．評価の結果，非同期式論理を用いたプロセッサの性能を同期式論理を用いた場合の性能に近づけることができた．

提案した Cascade ALU アーキテクチャの性能をシミュレーションにより評価した結果，Cascade ALU アーキテクチャの性能が ALU 遅延のみの削減で大幅に向上し，ある程度 ALU 遅延を削減できれば，Cascade ALU アーキテクチャで out-of-order アーキテクチャと同等の性能を達成できることがわかった．また，同等の性能となる条件において，分岐予測ミスに伴い破棄される命令数を少なくできることから，Cascade ALU アーキテクチャは消費電力の削減にも効果的であることもわかった．さらに，ALU 間配線の利用状況の評価から，連続した命令が割り当てられる ALU をできるだけ近くに配置することで，同時発行幅の増加による ALU 間配線の遅延増加を抑制できることがわかった．

また，Cascade ALU アーキテクチャの実装を行った結果，現行の 0.25 μm プロセスを用いた 4 命令同時発行の Cascade ALU アーキテクチャの性能が，out-of-order アーキテクチャを用いている MIPS R10000 に近いものであることがわかった．また，実装した Cascade ALU アーキテクチャの面積は，MIPS R10000 より小さくなることがわかった．

これらの結果から，Cascade ALU アーキテクチャを用いることで，プロセス微細化で配線遅延が減少しないことによるクロック周波数の制約を解決できることが示された．しかし，Cascade ALU アーキテクチャでは，アプリケーションが持つ IPC の限界を解決することはできない．そのため，Cascade ALU アーキテクチャに基づいて，近年提案されている SMT アーキテクチャのような IPC の限界を解決するアーキテクチャを検討する必要があることがわかった．