

非同期式カスケード ALU アーキテクチャの提案

3N-5

上野 洋一郎†

深作 泉†

中村 宏†

南谷 崇†

†東京工業大学 情報理工学研究所

†東京大学 先端科学技術研究センター

1 はじめに

近年のデバイス技術の進歩により次々と高速な半導体論理素子が開発されている。しかし半導体の高速化に伴って遅延全体の中で配線遅延の占める割合が増加し、回路全体にクロックを分配する必要のある同期式回路では素子の高速性を十分に生かすことが困難になりつつある [1]。このような非常に高速な素子を生かす方法の一つとしてクロック信号の無い非同期式回路を用いる方法がある。

非同期式回路によりプロセッサを実現する場合も同期式と同様に、プロセッサのアーキテクチャは性能を決定する重要な要素である。高速な非同期式プロセッサを実現するには、同期式システムで用いられる高速化技術に加えて、非同期式の特徴を生かすことの出来るアーキテクチャの導入が必要不可欠である。

本稿では非同期式プロセッサに適用することで高性能が達成できるカスケード ALU アーキテクチャを提案する。

2 非同期式回路

非同期式回路では、クロック信号がないため、演算の開始を要求する要求信号と演算の完了を通知する応答信号を用いる。図 1 の構成を例に要求信号と応答信号の機能を説明する。まず制御回路は動作させたい演算回路に対して要求信号を送る。演算回路は要求信号を受け取ると演算を開始し、演算が終了すると応答信号を返す。制御回路は応答信号が返ってきたことで演算結果が利用可能になったことを知り、次の処理を開始する。この方式を要求応答方式と呼ぶ。

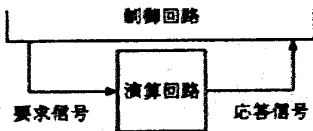


図 1: 要求応答方式

要求応答方式により、非同期式回路では演算回路の処理が終了とすぐ次の演算に移ることが出来る。したがって非同期式回路は演算の種類やデータによる演算時間の変動を利用して高速化できる。例えば ALU で論理演算を行なう場合と、0-1 という減算を行なう場合を比較すれば明らかに論理演算の方が高速である。この演算時間の変動は、同期式の場合には図 2 の上のように処理時間がクロックの幅で丸め込まれるので待ち時間になってしまうが、要求応答方式を用いる非同期式では図 2 の下に示すように演算を次々と行えるので、無駄なく高速に処理を進められる。

3 既存の高速化アーキテクチャ

既存の高速化アーキテクチャを単純に非同期式プロセッサに導入しても図 2 の非同期式の利点を必ずしも生かせない。

まず基本的なパイプラインについて考える。例えば、一つの命令の処理を図 3 のように、命令のフェッチ (IF)、デコード (ID)、実行 (EX)、レジスタファイルへの書き戻し

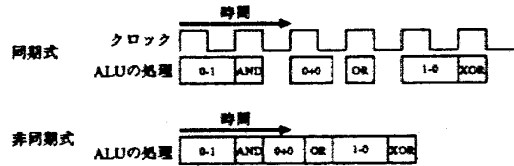


図 2: 同期式と非同期式での演算時間の変動

(WB) の 4 つのステージに分割したとすると、パイプライン化されたプロセッサでは図 4 に示すように複数の命令のステージをずらしながら処理する。



図 3: パイプラインのステージ分割例

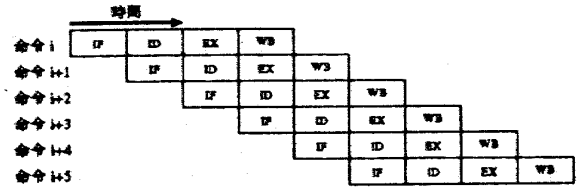


図 4: パイプラインプロセッサの命令処理

パイプラインを非同期式プロセッサに導入して EX ステージの演算が高速になっても、図 5 のようにその他のステージの処理が EX ステージでの演算よりも遅ければ、他のステージが律速段階となってしまい、速度は向上しない。

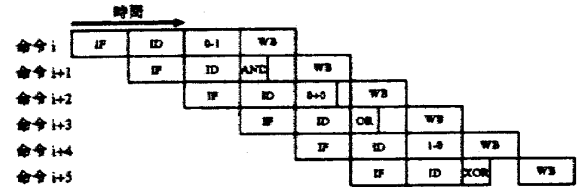


図 5: 非同期式パイプラインと演算時間の変動

次にパイプラインに加えて命令の並列処理を目指したスーパスカラと VLIW (Very Long Instruction Word) について考える。この 2 つはいずれも複数の命令処理パイプラインを用意し、依存関係のない複数の命令の同一ステージを同時に実行するというものである。図 4 に示すパイプライン処理を 2 つの命令を同時に処理できるスーパスカラプロセッサや VLIW プロセッサで理想的に実行した場合図 6 のようになる。スーパスカラと VLIW の主な違いは、複数ある ALU への命令の振り分けを決定するスケジューリングが命令の実行時か、コンパイル時かという点にある。

このスーパスカラや VLIW を単純に非同期化した場合も、EX ステージでの処理時間が同時実行している複数の命令の中の最も遅い命令で決定するのはもちろんのこと、パイプラインプロセッサである以上他のステージが律速段階となってしまえば非同期の利点を引き出すことができない。

4 カスケード ALU アーキテクチャ

スーパスカラや VLIW と同等な命令の並列処理能力を持ちながら図 2 に示す非同期式の利点を生かせるカスケード ALU アーキテクチャを提案する。

Proposal of Asynchronous Cascade ALU Architecture
Yoichiro Ueno, Izumi Fukasaku
Tokyo Institute of Technology
Hiroshi Nakamra, Takashi Nanya
University of Tokyo, Research Center for Advanced Science and Technology

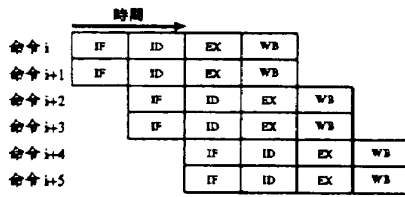


図 6: スーパスカラプロセッサ又は VLIW プロセッサの命令処理

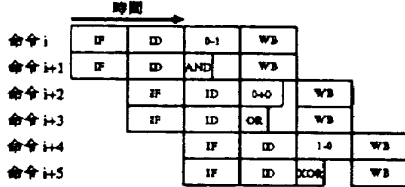


図 7: 非同期式スーパスカラ・VLIW と演算時間の変動

カスケード ALU とは、複数の ALU を様々な接続関係で繋げるようにデータ転送路を用意した EX ステージの構成法である。例として 2 個の ALU を使ったカスケード ALU の構成を図 8 に示す。この図の 2 つの ALU のうち、ALU1 で実行される命令は ALU2 で実行される命令よりも先行する命令である必要があるが、この条件が守られれば 2 つの命令の間にある任意の依存関係を実現できる。

例えば、2 つの命令にオペランドの依存関係が無ければ図 9 の左のように 2 つの ALU は独立の処理ができる。後続の命令の一つのオペランドにのみ依存関係が存在すれば図 9 の中央のように ALU1 の結果を ALU2 の入力の方に渡すことで処理できる。後続の命令の両方のオペランドに依存関係が存在しても図 9 の右のように ALU1 の結果を ALU2 の両方の入力に渡すことで処理できる。

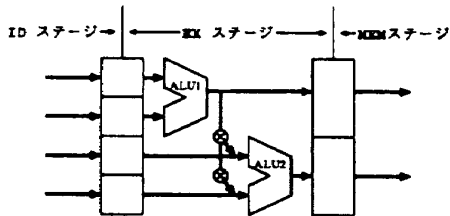


図 8: 2 個の ALU によるカスケード ALU

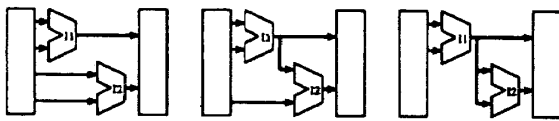


図 9: 2 個のカスケード ALU で可能な接続関係

このアーキテクチャの特徴は、演算の従属接続による高速化と、パイプライン化した場合に EX ステージを律速段階にすることによる高速化の 2 点である。

まず従属接続によって高速化できる例として積和命令が良く知られているが、依存関係を持つ連続した加減算でも非同期式ならば高速化が可能である。性能評価シミュレーションのために設計したゲートレベルの回路では、

[1つの加算の最悪遅延]: [2つの従属接続した加算の最悪遅延]

の比が Ripple Carry Adder では 1.00 : 1.13, Carry Look ahead Adder でも 1.00 : 1.51 という結果を得ている。

もう一つの EX ステージを律速段階にすることで高速化するという特徴は、同期式プロセッサが全ステージの処理時間を均等にして高速化していることと相反するように思える。しかしプロセッサ本来の目的は EX ステージで休む

ことなく演算し続けることにあるので、他のステージが EX ステージの足を引っ張ることが無ければ性能低下は招かない。例として前節でパイプライン化しても処理性能向上が見込めなかった図 5 の命令列をスーパスカラ/VLIW で実行した場合とカスケード ALU で実行した場合を考える。もし全ての命令間に依存関係がないと仮定するとスーパスカラ/VLIW の場合もカスケード ALU の場合も図 7 のようになる。もし全ての先行命令から後続命令への依存関係があると仮定すると、スーパスカラ/VLIW で実行した場合、図 10 に示すように EX ステージで待ち時間が発生する。しかしカスケード ALU では常に 2 命令を同時に投入するので図 10 の下の流れのように EX ステージが律速段階となるが、最終的な性能は向上している。

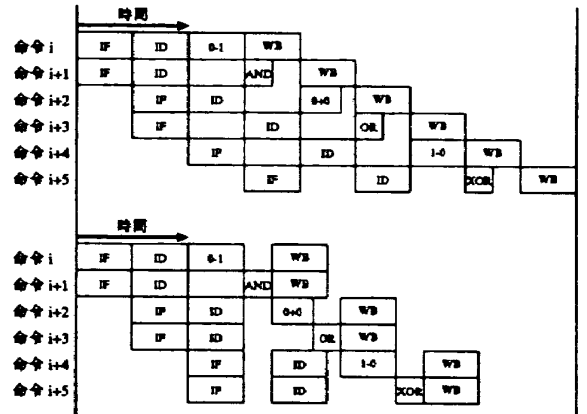


図 10: EX ステージを律速段階にすることで速度向上

5 まとめ

依存関係のない命令は並列に、依存関係のある命令は直列に実行して高性能を実現するアーキテクチャとしてカスケード ALU アーキテクチャを提案した。

カスケード ALU アーキテクチャは同期式プロセッサにも適用できるが、任意の複数命令を組み合わせた場合の演算時間の予測が困難であるのと、クロック信号による待ち時間が避けられないため本来の性能を發揮できない。

それに対して非同期式プロセッサでは、複数命令を組み合わせた場合の演算時間は演算回路が応答信号を返すことで知ることができ、演算時間も任意で、演算が終り次第の処理に移行できるので待ち時間が少ない。更に EX ステージを律速段階にすることにより高速化できる。以上の点から、カスケード ALU アーキテクチャは非同期式プロセッサに適したアーキテクチャと言える。

本アーキテクチャに関するより詳細な評価結果は [3] に述べられている。分岐命令などの実行制御命令も考慮した高速化が今後の課題である。

なお、本研究の一部は科研費補助金基盤研究 (B)09480049、及び (株) 半導体理工学研究センターとの共同研究によるものである。

参考文献

- [1] 南谷崇. 非同期式プロセッサ — 超高速 VLSI システムを目指して —. 情報処理, Vol. 34, No. 1, pp. 72-80, January 1993.
- [2] John L. Hennessy and David A. Patterson. *Computer Architecture A Quantitative Approach*. Morgan Kaufmann, 1990.
- [3] 深作泉, 上野洋一郎, 中村宏, 南谷崇. 非同期式カスケード ALU プロセッサの評価. 情報処理学会第 56 回全国大会, March 1998.