3H－3

VLDPにおけるデータ書き戻し削減手法
－Last Use検出によりデータ書き戻しを制御する方式－

田中　洋介、坂井修一、田中英彦
東京大学大学院工学系研究科

1. はじめに

デバイス技術の向上により１つのプロセッサで利用することができるトランジスタ数は増え続けている。プロセッサの性能向上に対する要求に応えるためには、将来これらのトランジスタをどのように利用するかがプロセッサアーキテクトの重要な課題となる。従来のスーパースカラ方式の単純な大規模化では、フォワーディング機構の複雑化によるクリティカルパスの増大、レジスタファイルアクセスの集中などがボトルネックとなり、性能向上に限界がある。

本稿では、（１）大規模化の可能性があるALU-Netについて説明し、（２）ALU-Netを利用することによりレジスタファイルアクセスの集中を緩和する方式を提案し、その可能性を検討する。

2. ALU-Net

ALU-Netとは数十から数百のALUが互いに最小限のスイッチとラッチを介してネットワーク状に結合された実行機構である[1]。ALU-Netは演算を行うだけでなく、レジスタファイルを介さないALU間のデータ転送(intra data access)、データ駆動的な演算発火制御なども行う。

ALU-Netには、動作遅延を削減できる、大規模化が可能、という利点がある。

ALU-Netを用いることによる動作遅延の削減には２つの異なる要因がある。第一点は、クリティカルパスになり得るフォワーディング機構を排除できるため、プロセッサ全体の動作周波数を向上できることである。第２点は、ALU-Net内の同期を非同期的にとることにより、ALU-Net内のクロックによる同期のオーバーヘッドを削減できる可能性があることである。

従来のスーパースカラ方式のフォワーディング機構を大規模化する際の問題点は主に３つある。まず、すべてのALU間にフォワーディングパスを張っているため、ハードウェアの複雑度がALUの数とreservation stationのエントリ数の積に比例することである。次に、フォワーディングパスがパイプラインステージを跨いで張られるため物理的に長いことである。そして最後に、非常に短期間のデータ待ち合わせのみにしか対応していないため、多くのデータ転送がレジスタを介して行われ、レジスタアクセスの集中がボトルネックとなることである。

これに対し、ALU-Netには次のような特長があるため、大規模化が可能である。まず、ネットワークの自由度を調節することにより、ALUの数の増大に伴うハードウェアの複雑度の増大を抑えることができる。さらに、ソフトウェア支援の利用などにより、データ依存解析ハードウェアの複雑度を下げられる可能性がある。次に、メッシュ構造などのように物理的に近いALU同士を接続するようなネットワークを考えることにより、intra data accessのパスを短く抑えることができる。そして最後に、待ち合わせ期間の柔軟性が高いintra data accessにより、レジスタファイルアクセスの集中を軽減できる。

ただし、ネットワークの自由度を下げるとintra data accessの割合が低下し、レジスタファイルアクセスの集中がボトルネックとなる懸念がある。このため、プログラムのデータ依存グラフの特徴を利用して、intra data accessの割合の低下を抑えつつハードウェアの複雑度を下げるネットワークの研究が必要である[2]。また、intra data accessによるデータ待ち合わせの期間が長くなると、ALUの保留時間が長くなりALUの構造依存がボトルネックとなる懸念がある。ALUの保留時間による、ALUの構造依存とレジスタアクセスの集中の緩和のトレードオフ関係の最適化についても研究が必要である。本稿では、このトレードオフ関係に特に注目する。

3. ALU deallocation on result's overwrite方式

ALU-Netでは、演算を完了したALUは演算結果を出力ラッチに保持したままの状態でその他のALUからのデータアクセスを受けることができる。このALUによる演算結果の保持時間はレジスタファイルポートとALUの構造依存の頻度に大きな影響を与える。

出力ラッチに演算結果を保持している期間、ALUは演算結果が上書きされるのを防ぐために次の演算を行うことができない。このため、演算結果の保持時間が長いと演算に利用できるALUの数が減少し、ALUの構造依存を増加させる。

ALU deallocation on result's overwrite scheme.
Yousuke Tanaka, Shuuichi Sakai, Hidehiko Tanaka
Graduate school of Engineering, The University of Tokyo

各データのレジスタファイルからの読み出し回数は、そのデータがALU-Net内に保持されていた間になされたintra data accessの回数だけ減少する。各データのレジスタファイルへの書き込みは、そのデータを生成したALUが解放される時に、そのデータに対するすべてのアクセスがintra data accessで完了したと保証される場合に省くことができる。

このように、演算結果の出力ラッチでの保持時間はALUとレジスタファイルポートの構造依存の頻度を大きく左右する。本稿では、インオーダALU割付、クロックによる同期を行う完全結合のALU-Netを仮定し、この保持時間の終了であるALUの解放のタイミングを

・演算完了、かつ
・論理destレジスタを上書きする命令に対するALU割付完了

を満たしたときとする方式（ALU deallocation on result's overwrite方式）を提案する。

インオーダALU割付、完全結合のALU-Netの場合、この条件が成立する時、当該データに対するすべてのアクセスがintra data accessで完結していると保証される。このため、ALU命令間のデータ転送がすべてintra data accessになるだけでなく、ALU命令間でのみ転送されるデータのレジスタファイルへの書き込みが不要となるので、レジスタファイルポートの構造依存は大幅に軽減されると考えられる。完全結合のネットワークでない場合は、ALU-Net内部のネットワークで結合されていないALU間のデータ転送に関するレジスタファイルアクセスの数だけレジスタファイルアクセスが増加する。

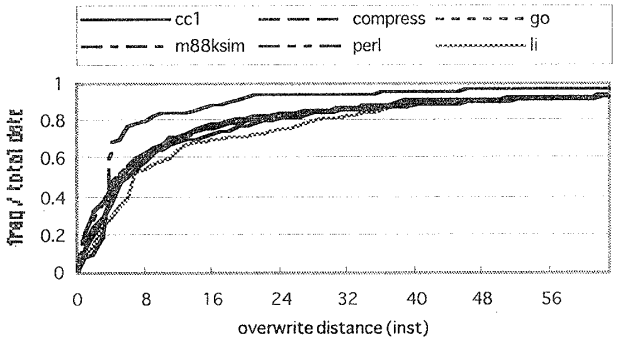

図1　データを生成する命令とそのデータを上書きする命令の間に実行された実行トレース上の命令数

ALU deallocation on result's overwrite方式はレジスタファイルポートの構造依存の軽減という面では理想的だが、長いALUの保留時間がALUの構造依存の頻発を引き起こす懸念がある。そこで、本方式におけるALUの保留時間の目安としてデータを生成する命令とそのデータを上書きする命令の間に実行された実行トレース上の命令数（overwrite distance）を調べた（図1）。評価にはSPECint95のベンチマークcc1、compress、go、m88ksim、perl、liを用いた。アーキテクチャはSPARC architecture version 8である。

図1よりoverwrite distanceは平均で14.5命令である。ALU deallocation on result's overwrite方式におけるALUの保留時間の目安はoverwrite distanceをIPC（Instructions Per Cycle）で割ることにより与えられるので、IPCが2、4、8の場合それぞれ7.3、3.6、1.8サイクルとなる。これはALUの構造依存がボトルネックとならない許容範囲内であると考えられる。ALUの解放が論理destレジスタを上書きする命令に対するALU割付でなく演算の完了によって駆動される場合は、実際のALU保留時間はこの見積もりよりも長くなる。しかし、演算が完了する前にALUを解放することはできないため、この場合のALU保留時間はALU deallocation on result's overwrite方式による副作用ではなく、プログラムのデータフローに依存する本質的なものであると考えられる。

ALU間でのみ転送されるデータの全データに対する割合の調査は今後の課題である。この割合はコンパイラサポートにより増大させることが可能だと考えられる。

4. まとめ

本稿ではALU-Netの利点について述べ、ALUの保留時間によるALUの構造依存とレジスタアクセスの集中の緩和のトレードオフ関係について述べた。

ALU deallocation on result's overwrite方式を提案し、インオーダ割り付け、クロック同期、完全結合のALU-Netを仮定すると、レジスタファイルアクセスの集中を緩和できることを述べた。本方式によるALU保留時間の目安としてoverwrite distanceを調べ、平均14.5命令程度であることが分かった。これはALUの構造依存がボトルネックとならない許容範囲であると考えられる。ALU間でのみ転送されるデータの全データに対する割合の調査は今後の課題である。

参考文献

1. 辻秀典，中村友洋，吉瀬謙二，安島雄一郎，高峰信，坂井修一，田中英彦．ALU-NET: VLDPアーキテクチャにおける命令実行機構．情報処理学会第57回全国大会講演論文集(1)，pp40．
2. 吉瀬謙二，中村友洋，辻秀典，安島雄一郎，田中英彦．ALU-Netを用いることによるデータ移動の効率化．情報処理学会第56回全国大会講演論文集(1)，pp109．