1Q−7 VLDP アーキテクチャの性能に関する初期的考察

中村 友洋, 吉瀬 謙二, 辻 秀典, 安島雄一郎, 高峰信, 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

筆者らは、21 世紀における大規模な VLSI を想定した次々世代のマイクロプロセッサ・アーキテクチャとして大規模データパス (VLDP :Very Large Data Path) アーキテクチャ[1] を提案している。本稿では、VLDP アーキテクチャの実現のために必要な技術及び、今後のデバイス技術等の動向を検討し、VLDP アーキテクチャの性能予測について定量的・定性的に述べる。

2 VLDP アーキテクチャ

VLDP アーキテクチャでは、本質的な演算部分を高速化することを第一の目的としている。そのために、演算処理部分である ALU を多数用意し、これを有効に活用するための機構をもたせるアプローチをとっている。つまり、多数の ALU を相互に接続 (ALU-NET) し、そこにデータパスを構築する実行方式である。これによって、プログラムの本質であるデータフローを ALU-NET 上にマップすることができ、オーバーヘッドの少ない処理が可能となる。ALU-NET を使って効率的に処理を行なうために、VLDP アーキテクチャは図 1のブロック図で示されるような構造としている。コントロールフロー先行展開で制御依存関係が解消された状態のデータパスを多数用意し、データパス先行展開で各パスごとにデータ依存関係に基づいてデータパスを作り、その ALU-NET 上へのマッピングを行なう。

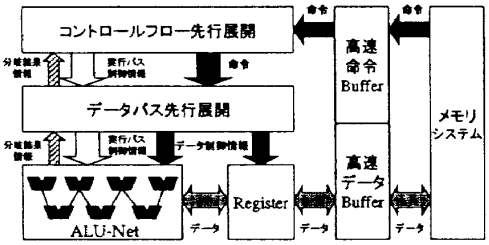

図 1: VLDP アーキテクチャ・ブロック図

3 問題点の考察・性能予測

3.1 メモリからの命令供給

多数の ALU を有効に活用するには、多数の命令を供給しなければならない。そのため従来のような 1 命令ごとにフェッチをする方式では、命令供給能力が不足する可能性が高い。この解決には、単純にキャッシュラインサイズを大きくするなどの方法も考えられるが、ミスアラインメントによる影響などがより顕著になり、必ずしも性能向上が得られるとは限らない。そこでトレースキャッシュの利用が有効であると考えられる。しかし、21 世紀の大規模化の規模を考えれば、よりアグレッシブな方式の検討が必要である。その 1 つの方式が命令ストリーミングである。命令ストリーミングとは、プロファイル情報などを利用して、複数のベーシックブロックにまたがる命令を 1 つのストリームにまとめることである。この際に、投機的な複数パスのストリーム化や、コンパイラによるストリーム情報の付加なども考えられる。命令ストリーミングにより 1 命令ごとのフェッチに比べてフェッチ回数が数十分の一 [4] になる効果が確認されている。命令ストリームングに関する詳細は、次の発表「命令ストリーミング: 複数パスの投機処理に適した命令列構成方式」で述べる。

3.2 コントロールフロー先行展開

コントロールフロー先行展開は、VLDP アーキテクチャにおけるフェッチ機構である。VLDP アーキテクチャでは、大規模なデータパス部分を使って命令実行を行なうため、命令の完了を待たずに投機的に命令フェッチをする必要がある。これは従来の分岐予測機構を使った Single Path の投機フェッチによって実現することもできるが、データパス部分の大規模化にともなって分岐予測性能の影響を大きく受け、大規模化には向かない。同様に図 2に示すすべてのパスをフェッチする Eager Fetch 方式も大規模化にともない不要な命令フェッチが爆発的に増加するため、コントロールフロー先行展開では、選択的な命令フェッチを行なう。フェッチパスの選択には文献 [3] で提案した multi BTAC と更新型 BHT の 2 つの分岐予測機構を設け、各コントロールフローごとに実行予測確率情報を管理する [4] ことで効率的な命令フェッチを実現する。図 3はコントロールフロー先行展開によって選択的にフェッチが行なわれたことを示すデータである。棒グラフ全体はすべてのパスをフェッチした場合であり、黒い部分が実行すべき命令数である。コントロールフロー先行展開では黒と白の部分の合計だけの命令フェッチに押えられている。その一方でフェッチされた

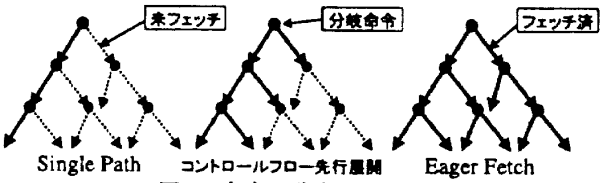

図 2: 命令の先行展開方式

Preliminary Studies of Very Large Data Path Architecture's Performance
Tomohiro NAKAMURA, Kenji KISE, Hidenori TSUJI, Yuichiro AJIMA, Makoto TAKAMINE, Shuichi SAKAI, Hidehiko TANAKA,
Graduate School of Engineering, University of Tokyo

命令列中の利用可能な並列度は図4に示すように、理想的な(分岐予測100 %)場合に近く得られており、動的な分岐予測による単一パスの投機フェッチに比べて大きく向上している。つまり極力不要な命令をフェッチせずに高い並列度を得ることが可能となっている。

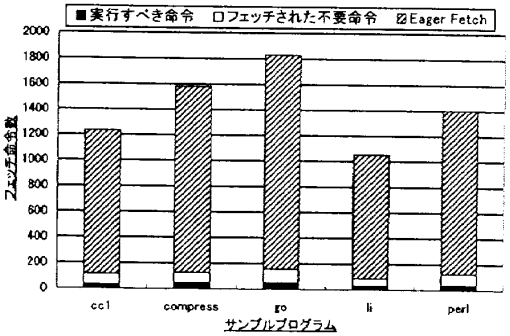

図 3: 選択的投機フェッチ

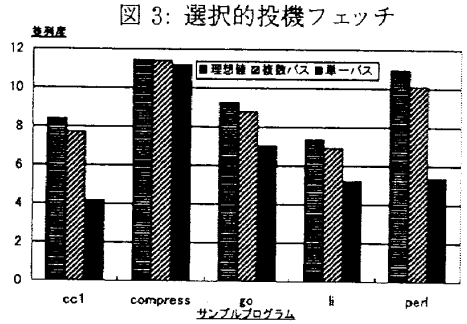

図 4: 利用可能な並列度

3.3 データパス先行展開

データパス先行展開は、データ依存解析を行ない、データ依存関係に基づいてALU-NET上に命令を割り付ける機構である。データ依存解析については既存のスーパースカラプロセッサなどでも行なわれているが、そのままの方式で大規模化を行なうことは命令ウィンドウの拡大と同じ問題となり難しい。そのために、例えばベーシックブロック単位、もしくは命令ストリーム単位で、各ブロック内のデータ依存解析を静的に行ない、その情報を付加することが1つの解決策である。またALU-NET上への命令のマッピングについては、データの存在(もしくは未出力の場合は出力先)を管理し、ALU-NET上のデータパス構築もしくはメモリシステムからのデータ転送として処理する。またALU-NETの資源管理も行なう必要がある。データパス先行展開における問題はこれらの機能を十分な処理速度で実現し、大規模化が可能な構造にすることである。詳細は、後の発表「VLDPアーキテクチャにおける実行パス制御方式の検討」で述べる。

3.4 ALU-NET

ALU-NETは演算器の相互結合で実現されている。この結合網はフォワーディングパスと機能的には同じ働きをしている。これを利用することで、データフローをハードウェア上にマップし、ここにデータを流すことで処理が進む。大規模な投機処理によって、命令間およびメモリとのデータ転送能力が問題になるが、ALU-NET上に多数の命令をマップしてしまえば、ローカルなデータ転送は集中型のレジスタ管理などをせずに各ALU間で行なわれる。このようなローカルなデータ転送をALU-NET内で吸収できる割合は図5に示すように、大規模化にともなって上がり、外部メモリとのデータ転送量増大を押えている。詳細は、後の発表「ALU-NET：VLDPアーキテクチャにおける命令実行機構」で述べる。

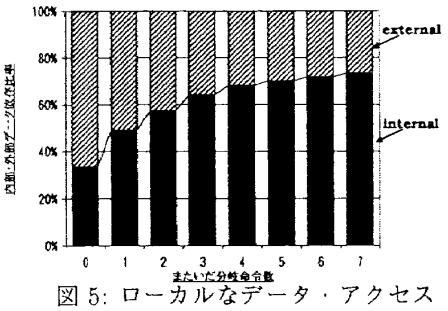

図 5: ローカルなデータ・アクセス

4 まとめ・全体性能予測

SIAのロードマップ[2]による2001年頃におけるデバイス技術予測などによれば、メモリからの命令転送能力を1024bit幅で1.5GHz動作と仮定して、最大8分岐命令をまたげるコントロールフロー先行展開のfetch能力をSPEC95intの各プログラムに対してシミュレーションした結果、14.3～45.9 G命令/sのフェッチ能力が確認できた。また図4からこのとき最大平均並列度は7～11にもなるので、実行部の動作周波数2～4GHz程度までは命令フェッチ能力が足りることが分かる。また図3からALU-NETの規模は200ALUs(10 × 20)程度が必要である。LOAD/STORE命令の実行頻度が20 %程度であり、図5から、その60 %程度がALU-NET上のデータ転送で吸収されるので、ALU-NET・メモリ間のデータ転送は1サイクル当たり1～2命令のロード命令を処理できる能力が最低限必要で、ALU-NET上に同時にマップされているロード命令の数は16命令程度である。以上の仮定をおけば、約10～40GOPS程度の演算性能が期待される。

謝辞

本研究の遂行にあたり，日本学術振興会の特別研究員制度(「プログラム解析に基づく次世代マイクロプロセッサアーキテクチャの研究」)のご支援を頂きました。感謝の意を表します。

参考文献

[1] 中村友洋, 吉瀬謙二, 辻秀典, 安島雄一郎, 田中英彦, "大規模データパスプロセッサの構想", 情報処理学会 計算機アーキテクチャ研究会, Vol.97, No.61, pp.13–18, 1997.

[2] "The National Technology Roadmap for Semiconductors", "Semiconductor Industry Association(SIA)", 1994.

[3] 中村友洋, 吉瀬謙二, 辻秀典, 安島雄一郎, 田中英彦. 大規模な投機的処理における分岐制御機構. 第56回 情処全国大会, Vol. 1, No. 5N-8, pp. 173–174, Mar 1998.

[4] 中村友洋, 吉瀬謙二, 辻秀典, 安島雄一郎, 高峰信, 田中英彦. 大規模データパスプロセッサにおける命令フェッチ機構. 電子情報通信学会集積回路研究会(ICD), コンピュータシステム研究会(CPSY), フォールトトレラントシステム研究会(FTS) 合同研究会, 信学技法, Vol. 98, No. 23, pp. 93–100, Apr 1998.