VLIWアーキテクチャにおける4並列浮動小数点演算の実現

4G－2

鳥島 剛、 山下 亮、 石川 禎
(株)東芝 情報処理・機器技術研究所

1.はじめに

高度産業用コンピュータVL2000シリーズは、産業用コンピュータとしては世界初のVLIWアーキテクチャを採用している。命令を1ワード(=4バイト)固定長とし、これを同時に4つずつ実行するものである。浮動小数点演算においても4命令同時実行を可能とする構造となっている。本稿ではその手法について述べる。

2.概要

VL2000シリーズの浮動小数点演算命令とその実行フィールドを図1に示す。フィールドとは、1語長の領域のことで、1フィールドに1命令を指定することができる。4倍語境界に置かれる4つのフィールドからなる領域を、クラスタと呼びこの単位で演算パイプラインに投入される。各フィールドの命令の割り当ては、既存のアプリケーションを調査し、頻度の高い加減算、型変換は、2フィールドで実行可能となっている。今回のインプリメントでは、1クラスタを2クロックでパイプライン処理することにより、レジスタファイルの出力ポート数を4ポートとして、論理を削減している。

浮動小数点演算命令は、その実行に複数クロックを要するものが多い。見かけ上1サイクルで実行を終了させるために、演算パイプラインを採用し、その実行終了をソフトウェアが意識しなくても済むように、演算結果の参照時点での待ち合わせの制御を実現した。

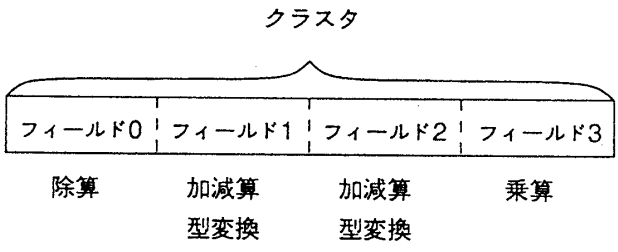

図1. 浮動小数点演算命令と命令フィールド

3.浮動小数点演算回路の構成

図2に浮動小数点演算回路の論理的な構成を示す。命令バスは、64ビット幅であるため、128ビットの命令を2回に分けてフェッチしている。オペランドリードは4ポート出力のレジスタファイルから2回に分けて行い、合計8オペランドを読み出す。4命令の結果とロードデータ、固定小数点レジスタからの転送データを書き込むために、レジスタファイルの入力は6ポートある。4命令を同時実行するために、それぞれのフィールドに対応した演算器を並列に配置している。

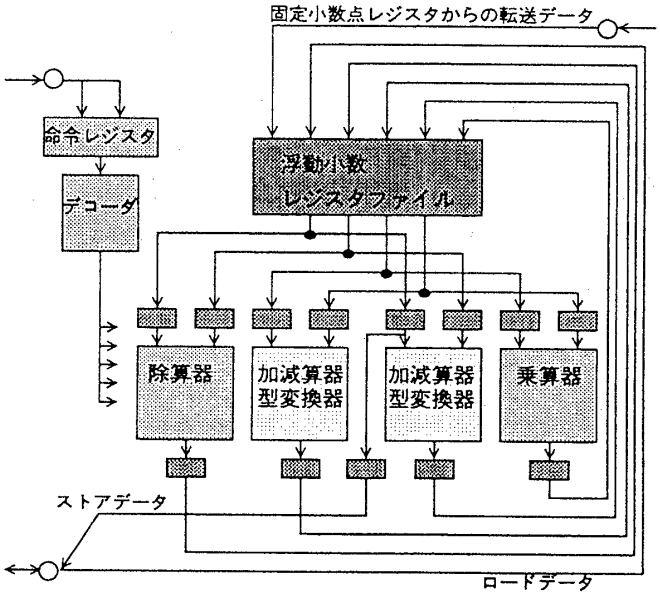

図2. 浮動小数点演算回路の構成

Implementation of 4-Parallel Floating-Point Calculation for VLIW Architecture.
Tsuyoshi TORISHIMA, Tadashi ISHIKAWA, Akira YAMASHITA
TOSHIBA CORPORATION

4.演算パイプラインと待ち合わせ制御

VL2000シリーズでは、浮動小数点演算命令の実行に演算パイプラインを用いている。これにより、演算の実行終了を待たずに、後続の命令を連続して実行することができる。実行に数クロックを要する浮動小数点演算命令であっても、見かけ上1サイクルで実行を完了する。また、演算パイプラインを用いることにより、同一クラスタに実行クロック数の異なる命令が配置された場合でも、各々の実行クロック数の違いは表面化することはない。図3にパイプライン処理の一例を示す。F、D、E、Wは命令パイプラインのステージを表し、順に命令フェッチ、デコード、実行、ライトバックの各ステージを表す。オペランドのフェッチはDステージで行う。E1～E5は演算パイプラインを表す。命令Aは実行に5クロックを要する浮動小数点演算命令とする。各ステージは2クロック(1パイプラインサイクル)ずつで処理されるが、Eステージは5クロックかけて演算が完了し、その後Wステージに移る。しかし、見かけ上はEステージは2クロックで実行を完了したように見えるため、命令AがまだEステージ上にある時でも、後続の命令がEステージに入ることができる。

命令の実行完了を待たずに後続の命令を投入することができるのは、命令の実行が終了する以前にデスティネーションレジスタを参照することがないとの条件の元でのみ、成立する。そこで問題になるのが、命令を実行している最中にその結果を参照した場合の制御である。命令の実行が終了する前にその結果を参照することを禁止する(そのようなオブジェクトコードを発生させない)という手段も考えられるが、その場合、ソフトウェアに対して制約を課すこととなる。ここでは、演算結果を参照した時点での待ち合わせ制御を取り入れた。演算パイプラインで実行中の命令のデスティネーションレジスタと後続の命令のソースレジスタとの依存関係を調べ、依存関係がある時は、後続の命令のオペランドフェッチを演算パイプラインで実行中の命令の実行が完了するまで、ハードウェアにより遅らせる制御をする。図4に演算の完了前に演算結果を参照した場合の例を示す。命令Bはレジスタの依存関係が無いため、2クロック毎に命令パイプラインを流れているが、命令Cは命令Aとレジスタの依存関係があるため、Dステージで3クロックを要している。この制御により、ソフトウェアは演算パイプラインの段数を意識する事なく、プログラミングができる。コンパイラは各命令の実行クロック数を意識し、この様な待ちが発生しない最適なオブジェクトコードを生成するようにしている。

5.まとめ

ＶＬＩＷアーキテクチャの計算機における演算パイプラインの実現と、待ち合わせ制御について述べた。この方式により、効率的な命令の実行が行われることが確認された。今後、さらに性能の向上をめざし、パイプラインの構造の見直し等を検討している。

参考文献

1)石川 禎、竹内 陽一郎
「VLIWアーキテクチャの実現」
情報処理学会第46回全国大会(1993)

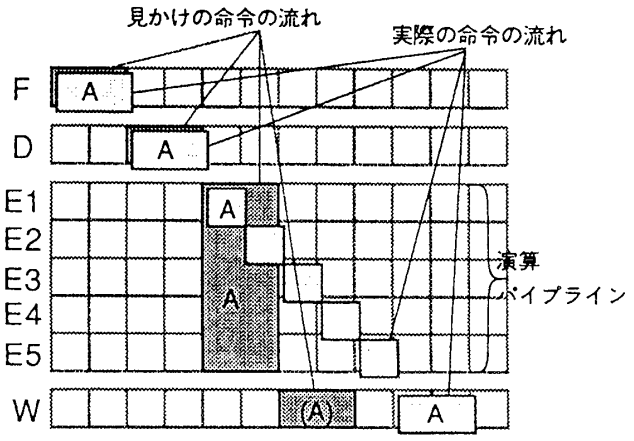

図3. パイプライン演算

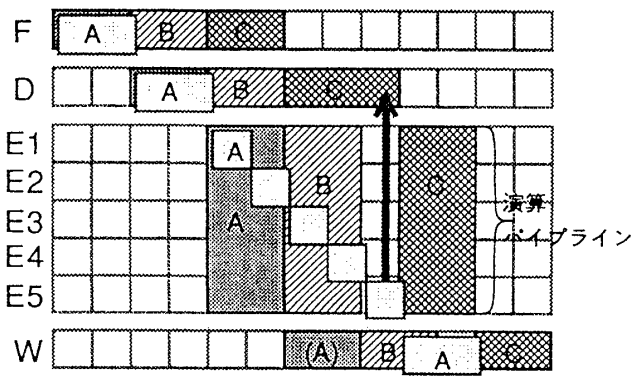

図4. 後続の命令でのデータ参照