

# VLIWアーキテクチャにおける浮動小数点演算の性能評価

1L-1

山下 亮 鳥島 剛 境 隆二 遠藤 浩太郎  
 (株) 東芝 情報・通信システム技術研究所

## 1. はじめに

高度産業用コンピュータVL2000シリーズでは新たにVLIWアーキテクチャを採用している。フィールドとよばれる1命令を指定する領域を1ワード(=4バイト)の固定長とし、4つのフィールドからなる4ワードの領域単位でパイプラインに投入し4命令を同時に実行するものである。

浮動小数点演算命令についても、ほかの命令と同様に4命令同時実行を可能とする構成をとっている。すべての浮動小数点演算命令をすべてのフィールドで実行可能とすることはハードウェアの論理量から不可能なため、各命令について実行可能なフィールドを限定することによってハードウェアの削減をおこなっている。4並列に実装した浮動小数点演算器は演算パイプラインを用いており、実行に複数サイクルかかる浮動小数点演算の実行サイクルを見かけ上1サイクルにすることによって演算の高速化を図っている。演算の制御方式については第47回全国大会にて報告した。

本稿では、性能測定結果をもとにVLIWアーキテクチャでの浮動小数点演算性能の評価結果について報告する。

## 2. 命令の概要

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。4つのフィールドからなる4ワードの領域をクラスタとよび、1クラスタの4つのフィールドをそれぞれフィールド0~3とよぶ。ハードウェアの論理量の増加を避けるためにフィールドによって指定することが可能な浮動小数点演算命令を限定している。

各フィールドの浮動小数点演算命令の割り当ては、既存のアプリケーションを調査し、実行頻度の高い加減算命令、型変換命令、0比較命令については2フィールドで実行可能、それ以外の乗算命令、除算命令については1フィールドで実行可能としている。

また、浮動小数点演算命令以外の命令についても実行フィールドを限定している命令があり(ロード、ストア命令など)、フィールド限定命令間での競合が少なくなるように実行フィールドを決定している。

## 3. 高速処理のための技術

VLIWアーキテクチャでは、命令の並列化スケジューリングはコンパイラでおこなうため、アプリケーションの性能はコンパイラの最適化技術によっても大きく左右される。このため、性能の向上にはハードウェアの高速化とあわせて最適化コンパイラの開発が必要不可欠である。

ハードウェアは、

- ・演算器の4並列実装
- ・演算パイプラインの採用
- ・演算結果参照時の待ち合わせ制御

などをおこなって演算処理の高速化を図っている。

待ち合わせ制御の採用などにより、実行される命令の依存関係によってはパイプラインハザードが増加し、演算パイプラインの効果が小さくなることもある。最適化コンパイラは、ハードウェアアーキテクチャに最適なコードを生成するようにハードウェアと並行して開発をおこなった。

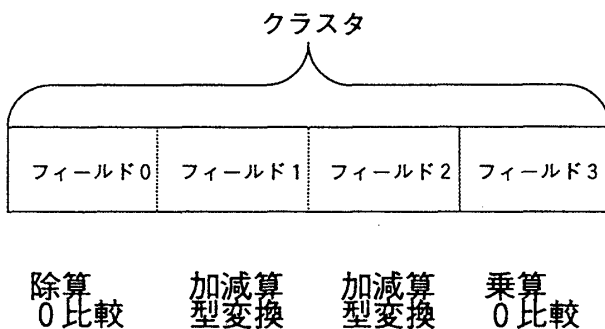


図1・浮動小数点演算命令と実行フィールド

Performance Evaluation of Floating-Point Calculation for VLIW Architecture

Akira YAMASHITA, Tsuyoshi TORISHIMA, Ryuuji SAKAI, Kohtaro ENDO

TOSHIBA CORPORATION

## 4. 性能測定

性能測定結果を図2に示す。

測定は、**Livemore Fortran Kernel, Whetstone** をもちいておこなった。比較のため、以下の3種のマシンの性能を測定し、周波数による正規化をおこなった。

- (1) 当社従来機  
(CISC)
- (2) VLIWアーキテクチャ1  
(RISC)
- (3) VLIWアーキテクチャ2  
(RISC)

(2)は、浮動小数点演算器がパイプライン化されておらず、演算器の一部を共有使用している。また、レジスタファイルの読みだしポートが4並列分備わっていない。このため、浮動小数点演算を並列実行する際にハードウェア資源競合によるパイプラインハザードが発生する構成になっている。

(3)は、演算パイプライン処理をおこなう演算器を4並列に実装しており、ハードウェア資源競合によるパイプラインハザードは発生しない構成となっている。

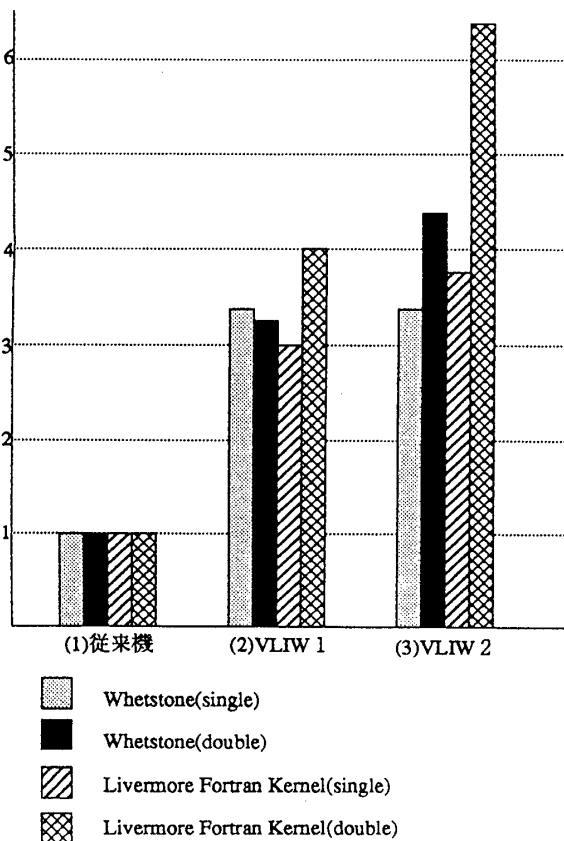


図2・性能測定結果

## 5. 考察

性能向上の要因としては、(1)から(2)での性能向上は、おもにCISCからVLIWへのアーキテクチャの変更と、最適化コンパイラの効果によるものである。また、(2)から(3)での性能向上は、おもに演算パイプラインの採用、演算器の4並列実装などハードウェア資源によるパイプラインハザードの削減によるものである。

(2)と(3)は、浮動小数点演算以外では同じ回路構成をとっているため、性能差は浮動小数点演算の頻度、1クラスタでの浮動小数点演算の並列度によって大きく変化する。

(1)、(2)のハードウェアは、単精度と倍精度の浮動小数点演算処理性能に差があるが、

(3)のハードウェアは除算をのぞいて精度による演算処理性能に差がなく、(1)、(2)と比較して倍精度の処理に優れている。このため単精度と比較して倍精度のほうが良い結果となっている。

**Whetstone** と比較して **Livemore Fortran Kernel** のほうが良い結果となっているが、これは命令の並列度が **Livemore Fortran Kernel** のほうが高かったためである。

今回の性能比較では、アーキテクチャ変更と最適化コンパイラによる性能向上が3倍以上、また、パイプラインハザードの削減による性能向上が倍精度で、1.3倍以上という結果を得ることができた。また、実際のアプリケーションによる性能測定でも同様の結果を得ている。

アーキテクチャ変更の効果が大きく、良い結果を得ることができたので、今後、動作周波数の高速化によるさらなる性能強化をめざしていく。

## 6. おわりに

VLIWアーキテクチャを採用しハードウェア、コンパイラともに新規開発となったが、当初目標である3倍をうわまわる性能を達成することができた。今後、さらなる性能向上を目指し、パイプライン構造等の見直し、周波数向上を検討している。

## 参考文献

- 1) 鳥島 剛、山下 亮、石川 禎  
「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」  
情報処理学会第47回全国大会(1993)
- 2) 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)