

3L-6 VLIW 計算機における条件実行制御と最適化コンパイラの実アプリケーションによる評価*

森本 展行、境 隆二、遠藤 浩太郎
 (株) 東芝 情報・通信システム技術研究所

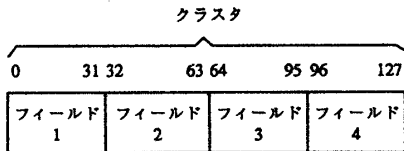
1 はじめに

VLIW 計算機は、並列 RISC アーキテクチャの1つであり、最適化コンパイラにより命令を並列にスケジュールすることで性能を上げる計算機である。ここで、並列化スケジュールリングの妨げとなるものに、条件分岐が挙げられる。我々が開発した VL2000 シリーズでは、条件分岐等の制御構造による依存関係を解消するために、条件実行制御アーキテクチャを採用している。基本的な条件実行制御の利用法としては、条件分岐命令の分岐先から命令を遅延スロットに持ってくる最適化と、ダイヤモンド型のフローグラフに適用する条件実行制御展開が挙げられる。実際、一般的なアプリケーションに対して、これらの最適化によって1割程度の性能向上が図れる [1]。

本稿では、この条件実行制御アーキテクチャを、さらに有効に活用するために、コンパイラ最適化処理の強化を行ない、幾つかの実アプリケーションに適用し評価を行なった結果について報告する。

2 アーキテクチャ

VL2000 シリーズでは、1ワード(32ビット)長の RISC 命令を4つのフィールドに指定して1つの VLIW 命令を構成している(図1)。分岐命令の遅延サイクルは1である。従って、遅延スロットは7フィールドあることになる(図2)。



フィールド: 1ワード長の命令指定領域
 VLIW命令: 4フィールドからなる同時に実行される命令の集まり
 クラスタ: 一つのVLIW命令

図1: VLIW 命令の構成

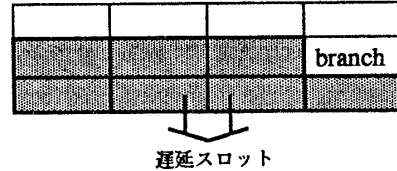


図2: 遅延スロット

3 条件実行制御方式

条件実行制御方式では直行性のある条件フラグを複数個もち、VLIW 命令を構成するすべてのフィールドに、条件フラグ番号(C-bit)と真偽値ビット(t/f)を指定する領域をもち、動的に命令の実行、非実行を制御する(図3)。

条件実行制御を行なうことによって、分岐命令を使わない命令列に展開する最適化が行なえる。この展開を条件実行制御展開と呼ぶ。

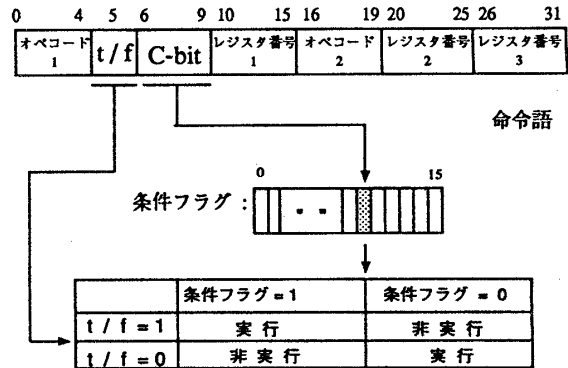


図3: 命令語に対する条件実行制御

4 評価アプリケーション

評価は、14種の実アプリケーションからなる SPECfp、6種の実アプリケーションからなる SPECint などに対しておこなった。SPECint に関しては、もともと分岐命令が多いアプリケーションで構成されているため効果がある。一方、SPECfp に関しては、科学技術計算であるので、条件実行制御展開の効果はあまりない。しかし、SPECfp の

*Evaluation of the conditional execution control and compiler optimization by some application programs on VLIW computer
 Nobuyuki Morimoto, Ryuji Sakai, Kotaro Endo
 Information & Communications Systems Laboratory
 TOSHIBA Corporation

mdljdp2、mdljsp2 においては、顕著に効果が見られた。以下、本報告では、mdljdp2 について説明する。

mdljdp2 は、分子の運動をシミュレーションする科学技術計算プログラムである。mdljdp2 のホットスポット (全実行時間に対して多くの時間を費している部分) は、分子間の距離を計算する部分である。このホットスポットでは、X 方向、Y 方向、Z 方向に対する計算が、それぞれ以下の if - then - else if - endif 型の条件文でインプリメントされている。

```
IF ( X .LT. - K ) THEN
    処理 b
ELSE IF ( X .GT. K ) THEN
    処理 c
END IF          ( K :定数 )
```

図 4: mdljdp2 のホットスポットの単位構造

5 条件実行制御と最適化

この構造を通常のフローグラフで表すと図 5 の左図のようになる。このような制御構造を持つ場合に、判定式の成立条件が排他的であるかどうかをコンパイラが判断し、最適化処理をほどこせば、フローグラフは図 5 の右図のように 3 並列に変形でき得る。このあと、条件実行制御展開が行をおこなえば、高並列スケジュールが可能となる。

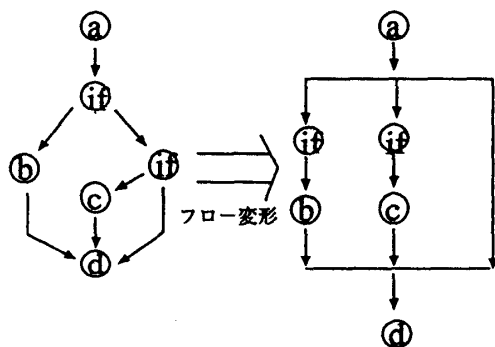


図 5: フロー変形

6 評価方法

VL2000 シリーズで実装されている並列化コンパイラの最適化レベル「-O2」を指定して生成したオブジェクトモジュールと最適化レベル「-O2」に加えて上記の最適化処理により生成したオブジェクトモジュールで評価を行った。

- ホットスポットのクラスタ数をカウントし nop フィールド (なにもしない命令) の割合を比較し、並列性を

見る。

- VL2000 シリーズ上で、異なる最適化を行なった 2 つのモジュールの実行性能を評価

7 評価結果

条件実行制御展開を行なうことによって高並列にスケジューリングされることがわかる。次に、「-O2」+「条件実行制御展開」の「-O2」に対する実行性能を評価した結果を表 1 に示す。

表 1: 評価結果

	-O2	-O2 + 条件実行制御展開
クラスタ数	72	44
nop の割合	50.1 %	29.0 %
SPEC 値比	1	1.25

条件実行制御展開を行なうことによって、並列性が向上し、また、実行性能も 25 % の向上が得られた。

8 おわりに

条件実行制御展開を積極的に適用することによって、性能向上が図れることが確認できた。これを行なうためには、分岐条件間の並列性の判定や不等式を扱う最適化の強化が必要である。また、条件分岐が大半を占めるような実アプリケーションでは、分岐のため命令キャッシュミスが多く発生するものもある。このような、アプリケーションに対して、条件実行制御展開を行なうことは、命令キャッシュミスの軽減にもつながり、性能向上が図れる可能性がある。一方、むやみに条件実行制御展開を行うことは、展開される命令数が多い時や条件成立側の命令数と非成立側の命令数のバランスが悪い時には返って性能を劣化させてしまう場合もある。今後、こういった問題点を解決していく予定である。

参考文献

- [1] 境 隆二、竹内 陽一郎、石川 禎
VLIW 計算機における条件実行アーキテクチャの評価とコンパイラの役割
情報処理学会第 48 回全国大会 (1994)