4F－9

ジオメトリプロセサ Procyon
－ソフトウェア・バイパス制御方式－

新井正樹　安里彰　小沢年弘
（株）富士通研究所

1　はじめに

ジオメトリプロセサ Procyon[1] はハードウェアを簡素化するために命令のソースオペランドの値をレジスタあるいは複数のバイパス出力のいずれから読むかをソフトウェアで指定するという特徴をもった4並列の VLIW プロセサである。Procyon では、各 VLIW 命令のソースオペランドの値をバイパスあるいはレジスタから読む方法を、パイプラインのタイミングを考慮してアセンブリコードで陽に指定する必要がある。本論文では、コンパイラによる Procyon のソフトウェア・バイパス制御方式について述べる。

2　ソフトウェア・バイパス制御方式

Procyon のアセンブリコードの例を図1に、その実行

```
(1): nop; add  %r2, %r0@R, %r1@R; nop; nop;
(2): nop; addi %r3, %r2@E, _A   ; nop; nop;
(3): nop; add  %r4, %r2@N, %r3@E; nop; nop;
```

図 1: Procyon のアセンブリコード例

時のパイプラインの流れを図2に示す。図1では最初のオペランドが出力レジスタを表す。整数系命令は D, E, N, W の4段のパイプラインをもち、E, N, W の各パイプラインステージで結果の値をバイパスに出力する。図1と図2で、VLIW 命令 (2) の %r2@E はレジスタ %r2 の値として VLIW 命令 (1) がバイパス E に出力した演算結果の値をデコードステージ D で使用することを表す。同様に、VLIW 命令 (3) の %r2@N はレジスタ %r2 の値として VLIW 命令 (1) がバイパス N に出力した演算結果の値を使用することを表す。VLIW 命令 (1) の %r0@R はレジスタ %r0 の値をバイパスではなく、レジスタから読むことを表す。

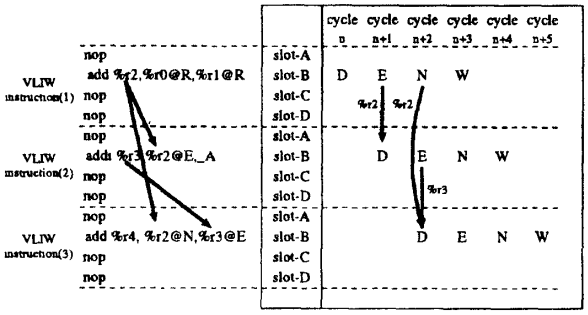

図 2: 実行時のパイプラインの流れ

3　コンパイラでの問題点

3.1　命令スケジューリング時のバイパスの扱い

Procyon のソフトウェア・バイパス制御方式のために、コンパイラによる命令スケジューリング時にどのようにバイパスについて考慮したらよいか、という問題が生じる。コンパイラでは一般に VLIW プロセサの命令スケジューリングを次のように行う。

1. 対象プログラムを VLIW エレメントの列に変換する。
2. VLIW エレメントの命令スケジューリングを行う。
3. 命令スケジューリングの結果から VLIW 命令を作成する。

例として、次の VLIW エレメントの列の命令スケジューリングを考える。

```
1: add  %r2, %r0, %r1
2: addi %r3, %r2, _A
3: add  %r4, %r2, %r3
```

命令スケジューリングには図3に示す資源予約表を使用する。一般の VLIW プロセサでは、資源予約表の書き込み部分には書き込みが終了するサイクルにエレメント番号を登録すれば良い。Procyon の場合には、資源予約表の書き込み部分には、エレメント番号だけでなくバイ

Geometry processor Procyon
- Software bypass control -
Masaki Arai, Akira Asato and Toshihiro Ozawa
FUJITSU LABORATORIES LTD.
4-1-1, Kamikodanaka Nakahara-ku, Kawasaki 211, Japan

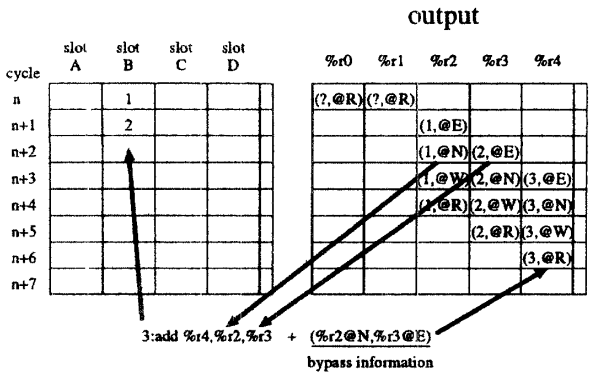

図 3: Procyon の資源予約表

バス情報も登録する。また、エレメントを資源予約表に登録するときに、どのレジスタ・バイパスからオペランドの値を読むかを命令に付加する。例えば、図3でエレメント３は、スロット B がサイクル n + 2 で空いていて、かつソースオペランド %r2 と %r3 をそれぞれバイパス N と E から読むことができるので、サイクルn + 2 のスロット B にスケジューリングできる。エレメント３を資源予約表に登録するとき、エレメント３と使用したバイパス情報 %r2@N と %r3@E を一緒に保存し、アセンブリコード生成時に使用する。この資源予約表から、結果として図1の VLIW 命令列を生成することができる。

3.2　制御フローグラフの合流点

Procyon では、制御フローグラフの合流点で、複数の先行基本ブロックが定義するレジスタの値を使用する命令のレジスタ・バイパス指定をどのように決定したらよいか、という問題が生じる。また、制御フローグラフは一般に有向循環グラフであるため、合流点の基本ブロックの命令スケジューリングを行う時に、先行基本ブロックの資源予約表を参照できない場合がある。我々のコンパイラでは、制御フローグラフの合流点の最適化のために、命令スケジューリング時に、制御フローグラフの合流点の基本ブロックの状態を次のふたつに場合分けして扱う。

3.2.1　先行基本ブロックの命令スケジューリングがすべて終了している場合

この場合には、先行基本ブロックが定義するレジスタの値をバイパスから読むために、先行基本ブロックの資源予約表を参照し、基本ブロックのスケジューリングを行う。基本ブロックに複数の先行基本ブロックが定義するレジスタの値を使用するエレメントが存在する場合には、可能ならば、先行基本ブロックの空きスロットにレジスタ間 move 命令を生成することで基本ブロックの入口でのバイパスのタイミングを合わせる。バイパスのタイミング調整の例を図4に示す。バイパスのタイミング

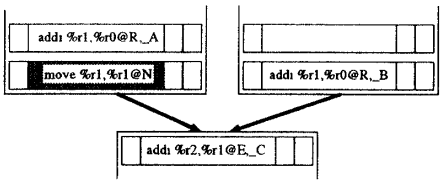

図 4: move 命令によるバイパスのタイミング調整

を合わせることができない場合には、オペランドの値をレジスタから読むようにスケジューリングを行う。

3.2.2　命令スケジューリングが終了していない先行基本ブロックが存在する場合

この場合には、命令スケジューリングが終了していない先行基本ブロック以前の命令によるレジスタへの書き込みが完了した後にそのレジスタの値を読む事を保証する必要がある。このために、命令スケジューリングが終了していない先行基本ブロック以前の基本ブロックが定義するオペランドを使用するエレメントは、レジスタへの書き込みが完了するサイクル以降に実行されるようにスケジューリングする。それ以外のレジスタに関しては 3.2.1 の方法でバイパスのタイミング調整を行う。

4　おわりに

コンパイラによるジオメトリプロセサ Procyon のソフトウェア・バイパス制御方式について述べた。今後はソフトウェア・バイパス制御方式を利用した広域命令スケジューリングについて検討して行く予定である。

日頃ご指導いただく和田英一顧問に深く感謝致します。

参考文献

[1] 安里彰 他. ジオメトリプロセサ *Procyon* －概要－, 第 55 回情報処理学会全国大会,1997.

[2] 西崎慎一郎 他. ジオメトリプロセサ *Procyon* －コンパイラ－, 第 55 回情報処理学会全国大会,1997.