4 F − 8

ジオメトリプロセサ Procyon
— コンパクション方式–

岩田 靖　安里 彰　細井 聡　西本 晴子　小沢 年弘
（株）富士通研究所

1 はじめに

VLIW 方式のプロセッサが実行する命令コードには，使用しない実行ユニットに対する nop 命令が必ず含まれる．これによりコードサイズが膨大になるという問題がある．

本稿では VLIW 方式のジオメトリプロセッサである Procyon で採用したロードモジュールのコンパクション方式について述べる．本方式では，命令フェッチ時に動的に復元可能な nop 命令をロードモジュール内のテキスト領域から取り除くことでコードサイズを圧縮するものである．

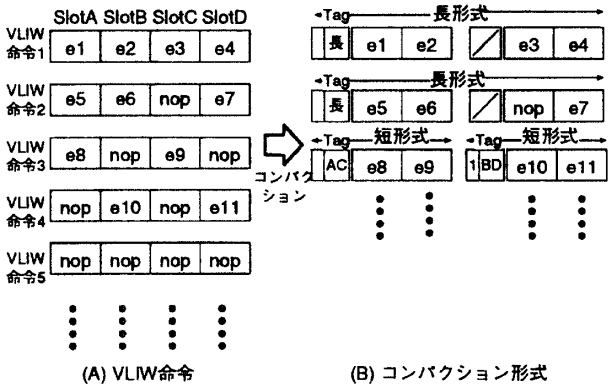

図 1: VLIW 命令とコンパクション

2 コンパクション方式

2.1 VLIW 命令

Procyon における 1 ワードの VLIW 命令は並列に実行される 4 個の命令エレメントから構成される．各命令エレメントは 30 ビットの長さを持つ．図 1(A) に Procyon の VLIW 方式の命令コードを示す．VLIW 命令内の各命令エレメントの位置をスロットと呼び，4 個のスロットをそれぞれ SlotA, SlotB, SlotC, SlotD と呼ぶ．各スロットの命令エレメントがアクセスできるハードウエアのリソースは決まっているので，命令エレメントを記述することができるスロットの位置は限られる．

2.2 長短形式と暗黙 nop

VLIW 方式の命令コードには，ある割合で nop 命令が必ず存在する．このような命令コードから nop 命令を省き，かつ命令実行時に動的に nop 命令を付加するコンパクション方式を実現する．

Procyon では，1 個の VLIW 命令に含まれる有効命令数が 3 個以上の場合は 4 個の命令エレメントで構成される長形式とよぶ命令形式でその VLIW 命令を表す．同様に 1 個の VLIW 命令において有効命令数が 2 個以下の場合は 2 個の命令エレメントで構成される短形式とよぶ命令形式でその VLIW 命令を表す．すなわち，短形式でひとつの VLIW 命令（4 並列）を表すことによりコードサイズの圧縮が行われる．図 1(B) に長形式と短形式で表された VLIW 命令の例を示す．図 1(B) で，e1～e11 は 30 ビットの命令エレメントを示し，Tag は当該命令が長形式なのか短形式なのか，短形式ならばどのスロットに有効命令が入るのかを表す情報フィールドを示している．この例では VLIW 命令 1 と 2 は有効命令が 3 個以上あるため長形式で表現し，VLIW 命令 3 と 4 は有効命令が 2 個以下であるため短形式で表現している．

Procyon では 2 個の命令エレメントに 4 ビットの Tag を付加した 64 ビット単位で命令コードが形成される．すなわち，長形式も短形式も全て 2 命令エレメント境界にアラインされる．短形式において，有効命令が入り得る場合の数は ${}_4C_2 = 6$ 通りある．長形式は 1 通りであるため，合計 7 通りの命令形式を表現できれば良い．そのため，Procyon では Tag の 4 ビットのうち 3 ビットで短形式の 7 通りの場合を表現している．残る 1 ビットは，後続の VLIW 命令が全て nop 命令で埋まっている場合に 1 を立てるものとする．このようにすることによって後続の 4 個の nop 命令を 1 ビットで表現

Geometry processor Procyon — Compaction method —
Yasushi Iwata, Akira Asato, Akira Hosoi,
Haruko Nishimoto and Toshihiro Ozawa
FUJITSU LABORATORIES LTD.

することができる．このように，1ビットで表された4個のnop命令を暗黙nopと呼ぶ．図1ではVLIW命令5が暗黙nopとしてTagの1ビットで表現されている．暗黙nopの概念は，Procyonの浮動小数点演算のレイテンシが2であることから，浮動小数点系の命令コードではnop命令が1行おきに入ることが予想されるために設けた機能である．

以上示したコンパクションは，アセンブラの機能の一部として行う．それに対して実行時には，コンパクションされている命令はフェッチステージにおいてTagの情報をもとに，動的にnop命令を挿入する．この操作はハードウエアにより行うことで命令を展開し実行する．

3　プログラムカウンタ

3.1　定義

命令圧縮を行うProcyonのプログラムカウンタ(PC)を次のように定義する．

「VLIW命令を長形式または短形式で表現し，その先頭部分すなわちTag領域の命令空間内アドレスをそのVLIW命令のPCとする．暗黙nopは命令空間内に存在しないのでPCはないものとする．」

すなわち，ProcyonにおいてはPCは圧縮されたコードに対して付加する．分岐命令の生成する分岐ターゲットアドレスはこのPCであることに注意する．

3.2　サブルーチンコール命令と戻り番地

サブルーチンコール命令は分岐時に戻り番地をある特別なレジスタに設定する必要がある．サブルーチンコールを含む分岐命令は全て1個の遅延スロットを持つため，リターン時には二つ先のVLIW命令から実行を開始するべきであり，従って戻り番地も二つ先のVLIW命令のPCになるべきである．命令圧縮を行っているため戻り番地はサブルーチンコールを含むVLIW命令と遅延スロットのVLIW命令の圧縮パターンによって二つ先のVLIW命令とのオフセット値が変化する．そこで，サブルーチンコール命令のコード内にリターンアドレスのオフセット(ROFF)を設けオフセット値を即値として与える．図2にサブルーチンコール命令の命令フォーマットを示す．戻り番地はROFFとサブルーチンコールを含むVLIW命令のPCとの和を計算してこれを特殊レジスタに保持する．

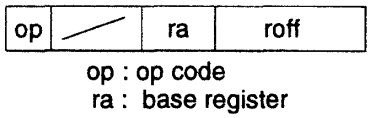

図2: サブルーチンコール命令の命令フォーマット

4　評価

ジオメトリ変換を行う実際のプログラムにおいて圧縮率を評価した．ここで用いているプログラムは，アセンブラ記述を人手で行ったものであり，コンパイラが出力したコードではない．図3においてその結果を示す．

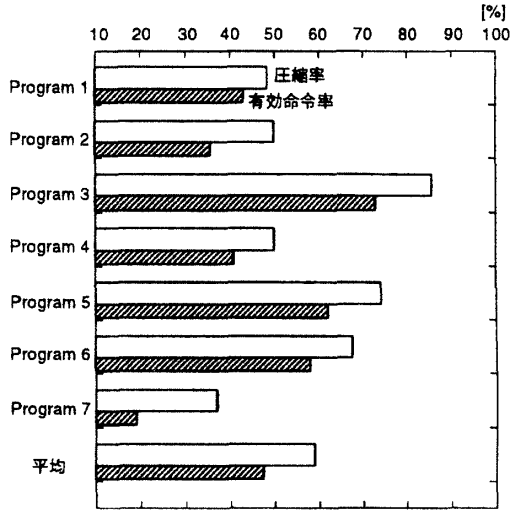

図3: 圧縮率と有効命令率

ここで圧縮率を次のように定義している．

$$\text{圧縮率} = \frac{\text{圧縮した命令コードサイズ}}{\text{圧縮しない命令コードサイズ}} \times 100[\%]$$

また，有効命令率とは全命令エレメントに対する有効命令エレメントの割合である．図3より，ここに示した7個のプログラムでは本稿で示したコンパクション方式により良好な圧縮が行われていることがわかる．

5　まとめ

本稿では，Procyonで採用した命令コードのコンパクション方式を示した．ジオメトリ変換を行ういくつかのプログラムにおいて圧縮率を評価したところ，良好な結果が達成されていることが明らかになった．