

5E-5

F O R T R A N 最適化の強化

—ベクトルプロセサ用レジスタ割当てにおけるループ分割方式—

鈴木幸男¹⁾ 佐藤 忍¹⁾ (日立ソフトウェアエンジニアリング)
 田中義一²⁾ (日立製作所中央研究所)
 青山明夫³⁾ (日立製作所ソフトウェア工場)

1. はじめに

スーパーコンピュータのアーキテクチャの中には、HITAC-S810のように、ベクトルプロセサ部は、スカラプロセサ部からベクトル命令をまとめて起動する方式がある。[1]この方式では、できるだけ「ベクトル処理単位」を大きくすることが性能上大きい効果があるが、ベクトル処理で使用するレジスタ類は有限であることから「ベクトル処理単位」にまとめられる範囲には限界がある。

従来のコンパイラでは、ベクトルレジスタ割当て以前にレジスタ必要量見積りによるループ分割を行っていた。そのため見積りに誤差が出たとき「ベクトル処理単位」が不当に小さくなったり、ベクトル処理をあきらめスカラ処理を行わなければならないことがあり、ベクトルプロセサの性能が十分引き出せないことがあった。

本報告ではベクトル化ループに対するベクトルレジスタ割当て時、レジスタ不足となった場合にループ分割を行うことが可能になったので報告する。

2. ベクトル化ループとベクトル処理単位

ベクトル命令とはベクトルプロセサで実行する加算、減算などの演算命令である。一連のベクトル命令をベクトル命令列といい、レジスタ割当て時のベクトル化ループはベクトル命令列の形で表現する。また、一回のベクトルプロセサの起動で実行するベクトル命令列をベクトル処理単位という。

一つのベクトル化ループが大きいとき、すなわち、ループ内のベクトル命令が多い場合には先に述べたようにレジスタ類が有限であることから、ベクトル命令列を分割しスカラプロセサ部からは複数回に分けてベクトルプロセサ部を起動する(図1)。

ここで、ベクトル化ループの分割点が最適になるようにベクトル処理単位を決定するには、ベクトルレジスタ割当て段階でレジスタが不足した場合にループ分割を行えばよい。ところが、ベクトルレジスタの内容を分割点の前後のベクトル処理単位で使用する場合、そのデータはレジスタ渡し可能なときとメモリ渡しとすべきときがある(図2)。メモリ渡しとするためにはメモリへの退避のために新たにメモリ(テンポラリベクトル領域)のアドレス付けのレジスタ等が必要になるので、レジスタが不足した時点のベクトル命令でループ分割することはできないことがある。

すなわち、ベクトル化ループの分割点はレジスタ不足のベクトル命令より遡り、既にベクトルレジスタを割当てた位置で分割しなければならないことがある。

[ソースプログラム]

```
DO 10 I=1,100
  A(I) = B(I) * C(I) ... ) ① 1回の起動でベクトル
  ;                               ) ② 処理可能な範囲
  D(I) = A(I) / E(I) ) ③
10 CONTINUE
```

[目的プログラム]

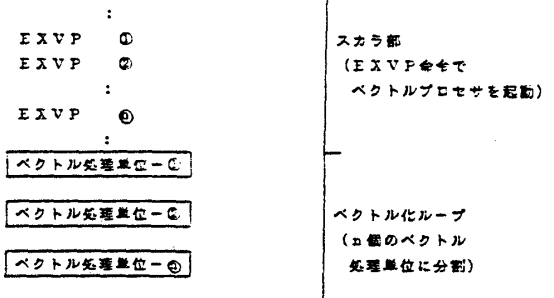


図1. ベクトル化ループの目的プログラム

本報告では、次の様なアーキテクチャを持つ計算機をモデルに考える。

- (1) ベクトルプロセサはスカラプロセサから起動される。
- (2) ループ不変なスカラデータは、ベクトルプロセサ起動時にスカラレジスタ(SR)にセットされる。
- (3) ベクトルデータは、ベクトルプロセサ起動時にそのアドレス情報がベクトルベースレジスタ(VBR)とベクトルアドレスレジスタ(VAIR)にセットされ、ベクトルプロセサ内でベクトルレジスタ(VR)を用いて、データの読み出し、演算、書き出しなどの処理が行われる。

Loop Splitting Method for Large Vectorized Loops.

1) Yukio SUZUKI 1) Shinobu SATOH 2) Yoshikazu TANAKA 3) Akio AOYAMA
 1) HITACHI SOFTWARE ENGINEERING Co., Ltd.
 2) Central Research Laboratory, HITACHI, Ltd.
 3) Software Works, HITACHI, Ltd.

```

DO 10 I=1, n
VRj = ...
      VRj = A(I) + B(I)
      = VRj
      レジスタ不足
      = VRj
10 CONTINUE

```

↓

メモリ渡し (n > ベクトルレジスタ要素数)

```

DO 10 I=1, n
      Vtmp = VRj
10 CONTINUE
DO 11 I=1, n
      VRj = Vtmp <
      = VRj
11 CONTINUE

```

レジスタ渡し (n < ベクトルレジスタ要素数)

```

DO 10 I=1, n
      VRj = ...
10 CONTINUE
DO 11 I=1, n
      = VRj <
11 CONTINUE

```

図2. ループ分割の例

3. ループ分割点の検出方法とループ分割方式

ベクトル化ループ内の各ベクトル命令に対するレジスタ割当ての過程で、ベクトルレジスタの割当て状況を一つのベクトル命令処理ごとに記録することによって、ループ分割点を検出する。

ベクトルレジスタ割当てとループ分割の過程を図3に示す。ここで、ループ分割点の検出には、以下のレジスタ割当て状況を使用すれば十分である。

- (1) ベクトルレジスタ割当てを行った所までの各種レジスタの使用量
- (2) ベクトルレジスタごとのレジスタの生死状態

すなわち、分割点となりうるかどうかは各ベクトル命令へのレジスタ割当てをした後、該ベクトル命令の直後で分割が起きた場合を仮定してベクトル処理で使用するレジスタを分割点以降でも使用するかどうかを調べる。また、それがレジスタ渡しとなるかメモリ渡しとなるかを判断して、分割のために新たに必要なレジスタの数を算出する。ここで得られたレジスタ数だけの未使用レジスタが残っていれば、該ベクトル命令の直後で分割可能であり、分割候補とする。

このように一つのベクトル命令に対応するレジスタ割当て時に分割可能点と、そのときのレジスタ割当て状況を記録しておき、レジスタ不足となったときに直前の分割可能点でループを分割する。分割処理後は分割点のベクトル命令からレジスタ割当てを再開する。

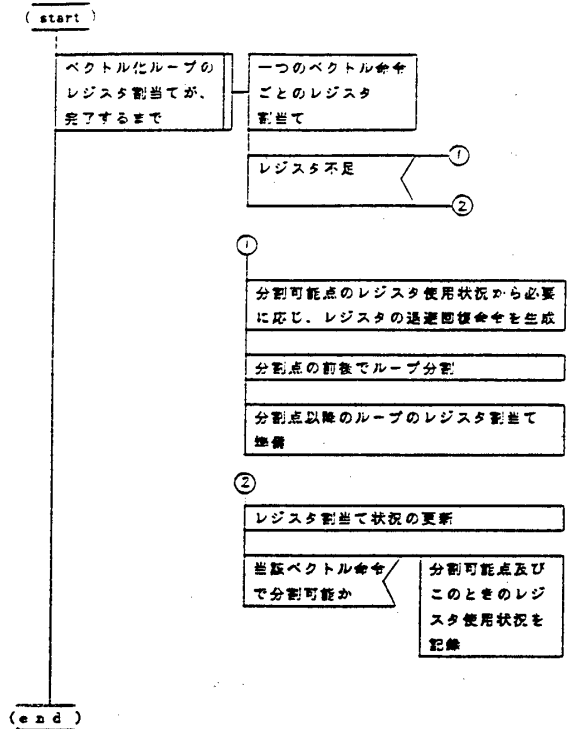


図3. ベクトルレジスタ割当てとループ分割の過程

4. おわりに

本報告で述べたベクトルプロセッサ用レジスタ割当てにおけるループ分割方式は、コンパイラへの組み込みが完了したところでありベクトル化ループの大きさにかかわらずループ分割を実現することができた。これにより、ベクトルプロセッサで実行する比率を向上させ、目的プログラムの実行時間の短縮が図れる。

参考文献

(1) 「スーパーコンピュータ HITAC S-810 FORTRAN コンパイラ」, 日立評論 vol. 65, no. 8 (1983), pp. 551-555